Automated Traffic Control for Smart Landing Facilities

by

Charles Henri Florin

Thesis submitted to the Faculty of Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Master of Science
In
Electrical Engineering

Committee members:
Dr. Timothy Pratt (Chairman)
Dr. Jeffrey Reed
Dr. Brian Woerner

December 6th, 2002
Blacksburg, VA

Keywords: SATS, SLF, TCAS, ADS-B, Secondary Radar, Mode S, PLL, ATC

Copyrights 2002, Charles H. Florin
Automated Traffic Control for Smart Landing Facilities

Charles Henri Florin

(ABSTRACT)

The Small Aircraft Transportation System (SATS) is a partnership between the FAA, the NASA, US aviation companies, universities and state and local aviation officials. The purpose of SATS is to develop a system to handle future increase in Air Traffic, reduce time-travel, develop automation in Air Traffic Control (ATC) and make better use of small aircraft and underused airports. The Smart Landing Facility (SLF) is an important part of the program. The SLF is a small airport upgraded with equipment to support SATS aircraft. Among the SLF equipment, SATS needs new detection equipment, and eventually automation.

This thesis investigates different techniques to avoid data collision in aircraft radar responses, and to reduce delays between landings and take offs. First, the paper shows how and when the radar receiver can separate two overlapped radar responses. Second, to avoid transponders responses overlapping, requirements in terms of aircraft safety distance are computed, different conflicts in air traffic around the SLF are examined and a solution is proposed for each case. And finally, the thesis investigates how far SATS can go in developing an automatic ATC system and what the role of future human operator will be in ATC.
Acknowledgments

I would like to begin by thanking my advisor and committee chairman, Dr. Timothy Pratt, for providing me this research subject, for all his advice and guidance. He is invaluable to the completion of this thesis. I would also like to thank Dr. Jeffrey Reed and Dr. Brian Woerner for serving on my committee.

I would like to thank Eric Shea for his guidance at the Center for Wireless Communications, and during this research. And finally, I would like to thank my family and friends for their support during my thesis and more generally during my work time at Virginia Tech.
Table of Contents

CHAPTER 1 – INTRODUCTION ...........................................................................1
  1.1 THE SATS PROGRAM .........................................................................................1
  1.2 RESEARCH OVERVIEW .....................................................................................2
  1.3 THESIS OUTLINE ............................................................................................3

CHAPTER 2 – RADAR SYSTEMS FOR SURVEILLANCE AND
COMMUNICATION WITH AIRCRAFT .................................................................5
  2.1 HISTORY OF RADAR (RADIO DETECTION AND RANGING) .....................6
  2.2 PRIMARY RADAR ..........................................................................................6
  2.3 IFF SYSTEMS ................................................................................................8
  2.4 SECONDARY SURVEILLANCE RADAR .........................................................9
    2.4.1 Secondary Surveillance Radar History ..............................................9
    2.4.2 SSR antenna .........................................................................................10
    2.4.3 Reply formats .......................................................................................12
  2.5 SSR LIMITATIONS .......................................................................................14
  2.6 TRAFFIC ADVISORY AND COLLISION AVOIDANCE SYSTEM, TCAS ..........15
  2.7 AUTOMATIC DEPENDENT SURVEILLANCE-BROADCAST (ADS-B) .........16
  2.8 SMALL AIRCRAFT TRANSPORTATION SYSTEM (SATS) PROGRAM ..........17

CHAPTER 3 – SEPARATION OF TWO OVERLAPPED RADAR
RESPONSES ............................................................................................................19
  INTRODUCTION .....................................................................................................19
  3.1 DESCRIPTION OF MODE C TRANSMISSION ..........................................21
  3.2 PHASE LOCK LOOPS (PLL) .........................................................................24
    3.2.1 PLL Basics ..........................................................................................24
## Table of Figures

| Figure 2-1 | Principle of radar systems ................................................................. | 7 |
| Figure 2-2 | Interrogation-signal formats ............................................................... | 11 |
| Figure 2-3 | Antenna interrogate beam and control patterns (Stevens, p.23) .................. | 12 |
| Figure 2-4 | Reply-signal formats, and an example (Stevens, p.25) ............................ | 13 |
| Figure 2-5 | Fruiting effect (Honold, p.52) ................................................................. | 14 |
| Figure 3-1 | A trapezoidal transponder pulse. Typical shape of demodulated pulses at the output of a SSR receiver ................................................................. | 21 |
| Figure 3-2 | Measured transponder reply frequency distribution (a) air-carrier aircraft (b) general aviation aircraft (Stevens) ................................................................. | 23 |
| Figure 3-3 | PLL block diagram (Kd is the multiplier gain, Kv is the VCO gain, Va is the multiplier output voltage, Ve is the error function at the filter output, and V0 is the Voltage Controlled Oscillator output voltage) ................................................................. | 24 |
| Figure 3-4 | Path along the phase detector characteristic. When the difference between the VCO output phase and the PLL input phase is p/2, the PLL locks ........................................................................................................... | 25 |
| Figure 3-5 | VCO output phase for a single sinusoid at the input and a critical damping factor ................................................................................................................................. | 27 |
| Figure 3-6 | PLL output for a single sinusoid at the input. VCO at the wrong frequency (VCO initial frequency 33Mhz is 10% higher than the received signal frequency equal to 30MHz) ................................................................................................................................. | 28 |
| Figure 3-7 | PLL output in a noisy environment S/N = 10dB for a noise equivalent bandwidth of 353 kHz ......................................................................................................................... | 30 |
| Figure 3-8 | One signal overlaps another in time ........................................................... | 31 |
| Figure 3-9 | PLL output for two overlapping signals at the same frequency, first signal starting at t = 0, second signal starting at t = 2ms ................................................................. | 32 |
Figure 3-10  PLL output for two overlapping sinusoids with different frequencies (First Signal with a frequency of 30MHz, starting at t=0, second signal with a frequency of 33MHz, starting at t=2ms)..................34

Figure 3-11  Block diagram of the demodulator for a single (not overlapped) transponder response. After frequency and phase detection, the received signal is multiplied with a sine wave corresponding to the same frequency and phase, before being filtered .................................................................36

Figure 3-12  Block diagram of the demodulator, separating two overlapped transponder responses. Once the first sequence is demodulated, it is subtracted from the received signal. The subtraction result is then demodulated the same way, and the second sequence is obtained. ...........37

Figure 3-13  IF received signal for two replies overlapped by 1.5ms (frequency difference of 20kHz, S/N = 15 dB)..................................................................................................................38

Figure 3-14  Recovery of the first signal, for two overlapped signals with different frequencies (f1=30MHz, f2=33MHz).................................................................39

Figure 3-15  Recovery of the second signal for two overlapped signals (f1=30MHz, f2=33MHz) ........................................................................................................40

Figure 3-16  Number of errors in the recovery of two overlapped signals with respect to the overlapping time (f1=30MHz, f2=33MHz).........................41

Figure 3-17  Details of the sum and difference beam patterns ..................42

Figure 3-18  Relative performance of the two demodulation methods for a frequency difference of 90kHz, f1=30MHz and f2=30.9MHz (number of errors with respect to the number of bits overlapped).........................44

Figure 3-19  Relative performance of the two demodulation methods for a frequency difference of 60kHz (number of errors with respect to the number of bits overlapped).................................................................45
Figure 3-20 Two consecutives demodulations for a frequency difference of 20kHz.................................................................................................................................................................47

Figure 3-21 two consecutive demodulations for a transition time of 0.06ms (a 60kHz frequency difference)..................................................................................................................................................................................48

Figure 3-22 two consecutives demodulations for a transition time of 0.10ms ........................................................................................................................................................................................................................................................................................................49

Figure 3-23 PLL output for a rise time of 0.45ms, non-delayed input ............50

Figure 3-24 PLL’s output with a 0.45 ms rise time pulse. The input is delayed by 0.6ms ........................................................................................................................................................................................................................................................................................................51

Figure 3-25 Separation results with a simulated phase error of 30% ..........52

Figure 3-26 3dB amplitude difference between the two signals (amplitude one for signal 1 and ½ for signal 2, with a 60kHz frequency difference). No error detected........................................................................................................................................................................................................................................................................................................53

Figure 3-27 10dB difference between the two signals amplitude (signal one amplitude equal to 1, signal two amplitude equal to 0.1, with a 60kHz frequency difference). Error occurs for the second (the weakest amplitude) signal........................................................................................................................................................................................................................................................................................................54

Figure 4-1 Two transponders pulses, coming from two aircraft 30m away from each other........................................................................................................................................................................................................................................................................................................61

Figure 4-2 Two different landing paths for SATS and current GPS approach ........................................................................................................................................................................................................................................................................................................62

Figure 4-3 GPS Approach plate for runway 12 at airport of Blacksburg, VA. ........................................................................................................................................................................................................................................................................................................66

Figure 4-4 An aircraft leaves Blacksburg airport and crosses the path of another aircraft arriving from Roanoke........................................................................................................................................................................................................................................................................................................67

Figure 4-5 (a) Initial Conditions: aircraft A and aircraft B at the same distance D........................................................................................................................................................................................................................................................................................................................................................................69

Figure 4-6 A third Aircraft C detects Aircraft A, and begins a holding pattern. Aircraft B proceeds while Aircraft A is holding. When Aircraft A finishes the
holding pattern, Aircraft A proceeds. When Aircraft C finishes the holding pattern, Aircraft C proceeds........................................................................................................70

Figure 4-7 A different approach for aircraft A........................................................71

Figure 4-8 Typical airport traffic pattern used by pilots for practice..............73

Figure 4-9 Airspace layout in Shea’s simulation..............................................74

Figure 4-10 Simulation of a new procedure for two aircraft heading to the IAF, one in front of the other, 3-D View of Air Traffic Pattern.................................76

Figure 4-11 Simulation of training aircraft with an arriving aircraft; 3-D view of the airport airspace ......................................................................................77

Figure 5-1 Decision process for traffic instructions.............................................85

Figure 5-2 Pilot radio announcements for approach in uncontrolled VFR conditions at an un-towered airport. Rectangle shows standard left hand traffic pattern for VFR arrivals and practice ..................................................89

Figure B-1 General PLL block diagram ............................................................124

Figure B-2 Demodulator block diagram ............................................................129

Figure B-3 Interrogation formats table ..............................................................131

Figure B-4 General response format from a Mode A transponder.................132
Chapter 1 – Introduction

The Smart Landing Facility (SLF) is an attempt to solve US Air Traffic congestion. This research explores Air Traffic Control Automation for small airports as a way to reduce delays in air travel. The goal is to coordinate traffic information, and to make a better use of current equipment. Suitably equipped aircraft would be able to fly in all weather conditions, with maximum safety, and minimum human Air Traffic Control.

1.1 The SATS program

According to a study by the National Transportation Safety Board (NTSB), there were 540 midair collisions involving general-aviation aircraft in the 20-year period, from 1960 to 1979 – an average of 27 per year or, about one every two weeks. Things have changed since those days, but the figures remain high: 11 to 15 per year in the 1990s.

No less than 490 of these collisions (90%) were between general aviation aircraft exclusively, 18 (3%) between general aviation and air-carrier aircraft, and 32 (6%) between general aviation and military aircraft.

Air traffic has dramatically increased since 1960s and this growth is not going to stop, despite the events of September 11th 2001. To solve what could become the 21st century transportation challenge, a national partnership, called Small Aircraft Transportation System (SATS), between NASA, the Department of Transportation, FAA, State governments, as well as the Department of Commerce & Industry, and universities has been created to focus on transportation system engineering, vehicle technologies, and enabling infrastructure technologies.
The main goal in this thesis is to design a Smart Landing Facility (SLF) for small airports. Most travel between two small cities needs to go through one of the thirty US big airports, the hubs, at present. As a consequence, these airports are regularly victims of a congested airspace. Small airports can provide shorter delays between departures and landings, allowing the traffic to be better spread over the US airspace.

The SLF is part of the SATS vision of air travel in year 2015, when light aircraft are fully equipped with Traffic Collision Avoidance System (TCAS) and a display showing topographic and traffic information. However, for the next twenty years, traffic using a smart landing facility will be a mixture of conventionally equipped GA aircraft with a Mode C transponder and no TCAS or SATS equipment. A SLF must support all conventional operations of the airport with a mixture of new services: meteorological information, topography, and traffic information.

This system will require new types of flight procedures. These new procedures will have to respect the capabilities of SATS and SLF equipment, and at the same time not be too different from what exists now. Otherwise, the FAA, pilots, and air traffic controllers will not accept them.

When such facilities are available, the demand for small aircraft and for pilots will increase. Therefore, we also have to make it easier to pilot small aircraft under both Instrumental Meteorological Conditions (IMC) and Visual Meteorological Conditions (VMC). Ideally, these aircraft will be self-separated from each other, and will not need the help of Air Traffic Control (ATC).

As a consequence, air travel between small airports using GA aircraft will be easier, safer, quicker and therefore less expensive.

1.2 Research Overview

In order to achieve the SATS goal, some form of automated air traffic is needed at un-towered airports. Today’s radar systems are too expensive for
small airports, and require many miles’ separation between two arriving or departing aircraft. Nevertheless, they are very efficient to detect aircraft and guide pilots. Thus, the first step toward automated air traffic is the design of a radar system that can be implemented with low cost at thousands of un-towered airports. The solution proposed in this thesis is the use of Traffic Collision Avoidance System (TCAS) as a ground sensor. The first part of the thesis is principally focused on a new method to separate two overlapped transponder responses, and the use of TCAS to reduce delays between two aircraft, and increase safety at small airports. The questions raised are: how can the system separate two overlapped transponders responses? In which cases can the system do that? And in which cases can the system fail to separate the two responses? A system to recover two overlapped transponders responses is designed and its limitations are demonstrated.

Moreover, reduced delays and new aircraft detection techniques for small airports will demand new procedures, which must be approved by the FAA. After studying new procedures for different cases, a simulation is done to validate our choices. It is also necessary to determine the limitations of the system, in terms of the number of movements allowed (aircraft landing or taking off) per hour. We will also identify in which cases the new system will not have a better detection capability than the current system.

Finally, the issues concerning a fully automated ATC, and different approaches for automation will be investigated. The decision process will be studied and analyzed. If an automated system is to be set up, one has to study the Human-Machine Interaction in ATC. It is necessary to determine how far one can go in automation without jeopardizing the safety of air travel.

1.3 Thesis outline

This thesis is divided into 6 chapters. Chapter 2 consists of an overview of air traffic surveillance techniques. After an historical overview, this chapter
describes Primary Radar, Identification Friend or Foe (IFF), Secondary Radar, Traffic Collision Avoidance System (TCAS) and Automated Dependent Surveillance-Broadcast (ADS-B). Chapter 3 reviews techniques for the separation of two overlapped received responses from Mode C transponders in a secondary radar receiver. A system to recover two overlapped signals is designed. To do so, a PLL and a separation algorithm are used. This technique is described theoretically, and then simulated. Then, the limitations and the performance of this system are explored. Chapter 4 examines the requirements to avoid transponders responses overlapping, new traffic procedures, taking into account the fact that SATS equipment gives traffic information to the pilots, and coordination between the pilots is possible. These procedures are described for different cases, and the limitations are established. As the question of a fully automated ATC system arises, Chapter 5 examines the limitations of full automation, in terms of Human-Machine Interaction, and proposes an automated system in which the roles of men and machine are balanced. Chapter 6 is a summary of this thesis, and a proposal for further investigations.
Radio Detection And Ranging (Radar) is used to track vehicles, and particularly aircraft. Surveillance radars were first developed for military purposes during World War II. Nowadays, radar is used worldwide for Air Traffic Control (ATC). Radars can be distinguished into two main categories: primary radar and secondary radar. To these two, another may be added: the Identification, Friend or Foe (IFF) system whose purposes are purely military. A primary radar interrogator (the radar on the ground) sends a pulse, which is reflected by a target (the aircraft). The sensor on the ground then detects this echo. With this system, the aircraft’s position is detected, but not its identity.

Secondary radar is not actually a true radar like the primary radar. A radar transmitter on the ground sends a pulse, called an interrogation, to an aircraft. The on-board transponder then replies with the identity of the aircraft and its altitude. Most aircraft (68% of General Aviation Aircraft, Smith & Baldwin 1994) are now equipped with a mode C transponders that replies with identity and altitude. Nevertheless, a small number of GA aircraft (18%) have no transponder. These aircraft are limited to Visual Flight Restrictions.

Air Surveillance is nowadays helped by the use of Global Positioning System (GPS). The Automatic Dependent Surveillance-Broadcast system (ADS-B) [Boisvert & Orlando, 1993] uses GPS to determine the location of the aircraft and broadcasts it. It also receives other aircraft ADS-B signals, and locates the aircraft that transmitted the signal.
2.1 History of radar (Radio Detection And Ranging)

In 1885, Thomas Edison wrote a patent for a system using radio waves to prevent collision at sea. After that, in 1920s and 1930s, the number of system using radio waves increased continuously. Some were used for altimetry, some others to sound off the ionosphere. It is also at that time that the very first experimentations dealing with waves reflection by diverse objects were made.

But the radar system, as we know it, was really born during World War II, for military purposes. To invade Great Britain, Hitler troops needed to force the British marine defense, which was well protected by the Royal Air Force. Therefore, the control of airspace was critical for both Germany and the Allies. Before radar, in 1940, anti-aircraft fire success rate was 1/1000 [5]. In 1945, at the end of the war, with the SCR-584 tracking radar to direct anti-aircraft fire and proximity fuses in shell, the ratio was 1/10. Moreover, after the war, radar was developed for civil purposes, and especially Air Traffic Control (ATC).

One can distinguish two kinds of radar systems. The first one is called Primary Radar, and works with the echo produced by any object reflecting a radio wave. The second kind is called Secondary Radar. With the current Secondary Radar system, the aircraft responds to a ground interrogator by sending information such as its altitude and its Identification Number (squawk code).

2.2 Primary Radar

Primary Radar works with a phenomenon called skin echo. When an electromagnetic wave is incident on an object, it partially reflects this wave. The
radar is nothing but a combination of a transmitter and a receiver, sending out a radio pulse, and listening for reflections (see Figure 2.1). If a reflection occurs, the radar determines the Angle of Arrival of the reflection, and computes the distance of the target by simply counting the delay between the transmission of the pulse and the reception of its reflection. This system can even give an approximation of the target’s nature (bird, aircraft, submarine, etc) according to its skin echo. In other words, as a submarine will not give the same reflection as an aircraft carrier, it is possible to distinguish them according to their wave reflections.

![Figure 2-1 Principle of radar systems](image)

As with any system, primary radar has good and bad points. The main feature of primary radar is its ability to detect small energy responses. During WWII, British radar services observed false responses due to raindrops. This ability made primary radar a good system for meteorology. Unfortunately, the system cannot distinguish two similar aircraft, because they give two similar responses. For example, Pearl Harbor’s tragic events occurred precisely because the radar system used at that time had been unable to tell if approaching aircraft were American or Japanese. The first system trying to overcome this problem was the British Identification Friend or Foe system (IFF), developed once again during WWII (see next section).

Moreover, primary radar provides a 2D location of the target (azimuth and range), but does not give any information about the target’s altitude. Secondary Surveillance radar overcomes this difficulty (see section 2.4).
2.3 IFF systems

The IFF system is the little sister of modern Secondary Surveillance Radar systems (SSR). Just before WWII, Britain’s Royal Air Force developed a surveillance system called Chain Home to detect German bombers. The IFF system was developed during the war. A ground-based transmitter broadcasts a radio signal, and any IFF equipped aircraft responds by broadcasting back a specific reply signal. Any aircraft that does not respond correctly can be quickly identified as a potential enemy. The system eventually operated in the 1030-1090 MHz band, now used by SSR.

The introduction of IFF systems needed time, and some pilots paid for its development with their life. At first, British pilots did not understand this new system, and some accidents occurred simply because they did not switch on the on-board transponder. But all early accidents were not due exclusively to pilots. In fact, in 1940, an officer working on the Thames Estuary erroneously identified a British aircraft as an enemy after having tracked it on the 180 degrees reciprocal bearing [Jerry Proc, 2002].

Several IFF systems were developed. F.C. Williams and Lord Bowden designed the two first systems, MKI and MKII, in 1940. In these systems, the on-board transponders were responding to ground-based Chain Home. Later, the Watson-Watt MKIII used a different interrogation in the frequency band 157-187 MHz. This separation of the primary radar’s pulse and IFF interrogation was a turning point, and nowadays primary and secondary radars work with separate interrogations. Because of a special response format, this system stopped having problems with sea or raindrops responses.

Later developments led to MK IV, using a higher frequency (G-band) and a narrower beam width (7-10 degrees). But, it was been quickly given up because a similar German system used the same frequency band. Therefore, German operators were able to detect Allied aircraft with the same precision. MK V, produced at the very end of the war, also called United Nations Beaconry (UNB),
improved the beam directivity and allowed different codes to distinguish one aircraft among several friendly airplanes. It used 12 different channels, in order to avoid jamming systems [Jerry Proc, 2002].

In the early 1950s, the United States developed MK X, whose frequency bands moved to 952-1139 MHz, and used twelve 17MHz channels. In 1952, nearly 50% of Navy ships were equipped with this system, and two years after, all of them were using this system. It used three modes, called General, Personal and Functional identification. Two logical ones, whose spacing depends on the mode used, composed each interrogation pulse. The transponder recognized which mode to use by computing the delay between the two logical ones. Spacing of 3 µs meant mode one was in use, 5 µs for mode two and 8 µs for mode three. Improvements in electronics and the use of cryptography led to MKXII, which is still in use nowadays, with some refinements.

As it is often the case, a military system was later used for civil purposes. These successive IFF systems permitted the development of civil Secondary Surveillance Radar, using the same technology.

2.4 Secondary Surveillance Radar

2.4.1 Secondary Surveillance Radar History

Secondary Surveillance Radar (SSR) is the adaptation of IFF systems to Air Traffic Control (ATC). During the 1960s, American airspace became so busy that it became difficult for air traffic controllers to distinguish aircraft among all the blips on the primary radar display. It became urgent to use a system that could recognize a single aircraft, and to provide each of them with a specific track in the airspace.

Secondary radar has many advantages over Primary radar:
- First, because the signal received at the ground station is produced by the aircraft transponder, the system range is a function of $1/R^2$, (where $R$ is the distance between the ground sensor and the aircraft). With Primary Radar, the same pulse is sent by the ground sensor and sent back by the aircraft. This single pulse goes through the same link twice. Therefore, the path loss is squared, and consequently Primary Radar range is a function of $1/R^4$.

- Second advantage: SSR not only detects aircraft, but also identifies them according to their code (called a squawk code). Aircraft respond to each interrogation by their squawk code or by sending back their altitude. Therefore, the airport ATC system is able to identify each target and to determine its altitude, which gives a 3-D location of the aircraft in the space.

- Third, SSR systems uses two different frequencies for interrogations (1030 MHz) and for replies (1090 MHz), avoiding any undesirable echoes from terrain or weather (called radar clutter), and allowing the use of different interrogators within a small area.

2.4.2 SSR antenna

Mode C SSR interrogator produces two different kinds of interrogations. One asks for aircraft ID, and the other for altitude. These two interrogations have to be different, and must be recognized by the aircraft’s transponder. Interrogations are divided into groups, called “modes”. As shown in Table 2.1, some of them are used for civilian and others for military purposes. Mode C is used almost exclusively by ATC in 2002.
The interrogation signal is composed of three pulses: P1, P2 and P3 (see Figure 2.2). The P1-P3 pulse spacing indicates which mode is used, and determines the reply format. The P2 pulse amplitude indicates in which precise direction the antenna beam is operating (see Figure 2.3). SSR antennas have a broad beam, around 5 degrees. Several methods exist to increase the directivity of the system [Honold, 1971]. The two-pulse method consists in feeding the P1 and P3 pulses to a directional antenna with high gain, and in feeding the P2 pulse to an omni-directional antenna, with low gain. Thus, if P2 pulse amplitude is smaller than the two others, the on-board transponder detects that the aircraft is within the main beam of the transmitting antenna. Otherwise, the aircraft is located on one of the radar antenna’s side lobes, and the interrogation is simply ignored.

![Figure 2-2](image)

**Figure 2-2** Interrogation-signal formats
Figure 2-3  Antenna interrogate beam and control patterns (Stevens, p.23)

Side lobes are very annoying for secondary radar. If an aircraft is located on one of these side lobes and responds to a radar interrogation, its azimuth will be displayed incorrectly.

2.4.3  Reply formats

Aircraft transponders reply with different format in response to each mode. Mode 1 uses 8 information bits, mode 2 and 3 use 12 information bits, and mode C uses 11 bits. To these information bits is added another bit, called X, always equal to a one, and located in the middle of the reply. Another pulse, called SPI, can be used if ATC requests it. To activate the bit, the pilot simply presses a
switch called IDENT on the transponder. The IDENT bit causes the aircraft 
alpha-numeric readout on the ATC radar display to go bright or to blink, so that 
the controller can immediately locate the aircraft on the display.

Figure 2-4  Reply-signal formats, and an example (Stevens, p.25)

The altitude information in a mode C reply is converted to an octal value 
ABCD (see Figure 2.4). For instance, the altimeter gives 34,800 ft. The altitude is 
the flight level 348. The decimal number 348 is then converted to the octal 
number 0534. In this example, A=0 B=5 C=3 D=4. The reply will be [F1=1, 1, 0, 
1, 0, 0, 0, X=1, 1, 0, 0, 0, 1, 1, F2=1].

In Mode C, the pulse D1 is not used. Therefore, only 2048 permutations are 
possible. This is sufficient to indicate height in 100 ft increment from –1000 ft to + 
121 000 ft. Since the altitude is obtained from a pressure transducer, it is 
necessary to calibrate the instrument to the local pressure.

The squawk code is an octal number (between 0000 and 7777) given to 
each aircraft when it is ready to take off. It is directly given by the reply signal 
ABCD, where A = A1+A2+A4 (from 0 to 7), and the same for B and C. Since D1 
is not used, certain permutations do not exist in Mode C.
2.5 SSR limitations

SSR has proved to be of very great value to Air Traffic Control (ATC), and is essential for the safety of the world’s air traffic. However, because of its success, SSR becomes less tolerant of shortcomings. Mainly two problems can occur: mutual interference effects and multipath phenomena.

When several secondary radars can see the same aircraft, mutual interference occurs. The fruiting effect (false replies unsynchronized in time) is related to the number of interrogators. If one single aircraft receives an interrogation, and replies to it, several secondary radars can receive this reply. As a consequence, all the radars that receive the response, but did not send the interrogation receive a false reply, and then may observe a false target (see Figure 2.5).

![Figure 2-5 Fruiting effect (Honold, p.52)](image)

The garbling effect is related to synchronized and unsynchronized reply code overlaps. Due to multipath phenomena, the reply can be reflected by buildings, trees and atmospheric conditions. Consequently, to the right reply is added the
same reply delayed in time and corrupted in amplitude. This second signal is likely to overlap the first one. Moreover, if another aircraft, close to the first one, replies to the same interrogation, the radar responses of the two aircraft will be overlapped. This happens especially for approaches, when a high number of aircraft are located in a small area. In other words, this happens only when it has the most disturbing effects.

In the next chapter, we will see how we can deal with this problem. Let us note that for multipath the second signal is not really important, since it is redundant, and with low amplitude compared to the first one. The system just has to extract the main response and throw the others away.

### 2.6 Traffic Advisory and Collision Avoidance System, TCAS

The Traffic Advisory and Collision Avoidance System is an airbone system using aircraft Mode S transponders to communicate avoidance decisions between aircraft. This system may appear under three forms:

- **TCAS I**: the first version of TCAS. It locates a threatening aircraft and gives its most likely direction. Pilots still have to locate the target visually.

- **TCAS II**: designed for larger air-carriers, this system has the same functionalities as TCAS I. Moreover, it determines the location of the threatening aircraft and a potential escape maneuver, and communicates it to the other aircraft, if it is TCAS II equipped. Nevertheless, this maneuver has to stay in the vertical plane.

- **TCAS III**: same functionalities than TCAS II, but the antenna provide information that allows a maneuver in the horizontal plane.
For light aircraft, the only available system is TCAS I. But with this system, the pilot cannot automatically locate the conflict, and has to determine the best escape maneuver. This method has two main disadvantages: first it allows human error (either in the location of the target visually, or in the escape maneuver), and second, the escape maneuver is likely to be known only by the pilot.

One solution of SATS consists in designing a new generation of TCAS, a TCAS IV, similar to TCAS III, but simple enough to be implemented on small aircraft, and at an acceptable cost.

Let us keep in mind that, since TCAS works with a mode S transponder, there is a necessity for the airports to get equipment compatible with the old Mode C transponders. Mode S transponders are compatible with Mode A and C, and TCAS used as a ground sensor is cheaper than a Secondary Radar system. A TCAS IV unit would therefore have two effects: to reduce the risk of air traffic collisions, and also to justify the installation of Smart Landing Facilities (SLF) at small airports [Stevens, 1988]. We will see in the next chapters how TCAS may be used as a ground sensor.

2.7 Automatic Dependent Surveillance-Broadcast (ADS-B)

The Automatic Dependent Surveillance-Broadcast system can be seen as a combination of TCAS and GPS. An ADS-B equipped aircraft broadcasts its position and squawk code or identification number through a Mode S transponder, and any aircraft or ATC facility can receive this information. This way, no one has to send an interrogation to know the exact location (GPS accuracy) and identity of any aircraft.

Even if this system makes the ATC task easier, and even if the sky is free of any interrogations, in congested airspace, the number of broadcasts becomes
enormous. Remember that with TCAS or SSR only a small fraction of aircraft, those present in the interrogator’s direction, will reply. With ADS-B, all aircraft are continuously broadcasting information sequences.

But, on the other hand, in congested airspace, a single aircraft can receive as many as 1000 interrogations per second, when TCAS is added to SSR. The use of ADS-B would simplify the task of the aircraft transponders, and set the airspace free of interrogations. Moreover, ADS-B transmission occurs at much lower rate than SSR replies. Therefore, the electromagnetic pollution, responsible for a large part of the noise floor in congested airspace, is reduced.

The FAA made demonstrations of ASD-B systems in 1995. One may distinguish three kinds of ADS-B systems: air-to-air, ground-to-ground, and air-to-ground. Air-to-Air systems can be viewed as a new generation of TCAS, detecting potential threatening aircraft, and proposing an escape maneuver. Ground-to-Ground is especially useful in large airports, where ATC and aircraft need to know the location of other aircraft and airport service vehicles. In 2001, a Boeing commercial aircraft crashed into a small Cessna while both of them were maneuvering on the ground in Milan airport. The ADS-B system allows ATC to know where each one of the vehicles is in any weather conditions. Finally, the Air-to-Ground application warns the pilot of any aircraft or vehicles close to the runway, and gives to the airport ATC the exact position and identity of the aircraft.

2.8 Small Aircraft Transportation System (SATS) program

The Small Aircraft Transportation System (SATS) is a partnership between NASA, the FAA, United States Aviation Industry, and universities. The program’s purpose is to develop a system to satisfy the increasing demand for safety of flight and a reduction of delays in the US air traffic.
Today, US air traffic is organized around 30 large airports (or hubs). Most of the time, passengers wanting to go from a small airport to another small airport are required to pass thought one of these hubs. The increasing demand for air transportation leads to the congestion of the airspace around these hubs. It is possible to fly straight from one small airport to another in Visual Flight Conditions (VFR) and Instrumental Meteorological Conditions (IMC). But SATS aims to improve the utility of un-towered airports by providing pilots of General Aviation Aircraft better information with which to fly between un-towered airports in IMC in greater safety.

Direct flights will increase the demand for air traffic, and therefore, the demand for more trained pilots, and more small aircraft. Many small airports are not equipped to face this increase, and a large number of aircraft operating around a small airport can become tricky. SATS also proposes to make single pilot operations easier and safer. The SLF will detect aircraft around the airport and broadcast information about the traffic, the weather and the terrain. It will also issue warning in case of traffic conflict.
Chapter 3 – Separation of two Overlapped Radar Responses

Introduction

Most of Air Traffic Control (ATC) near large airports is organized around a Secondary Surveillance Radar (SSR) and Mode C transponders on aircraft. The SSR sends an interrogation to an on-board transponder. This transponder (Mode C) answers with the ID and altitude of the aircraft on successive interrogations. The SSR computes the transponder signal's angle of arrival and the delay between the transmission of the interrogation sequence and the reception of the transponder reply, allowing the SSR to determine the azimuth and the location of the aircraft.

When two aircraft respond to the same radar's interrogation and are close to each other, their responses may overlap in time. In that case, the most sophisticated Secondary Surveillance Radars will simply extract information about the existence of targets, but will not identify them. This may be a problem if the ATC controller has not seen them for a long time. The solution adopted today is to instruct pilots to maintain a certain distance between their aircraft. If we need to increase airport capacity, we need to decrease this distance as much as we can. And since a radical change in today’s radar equipment is not an option for the FAA, the system has to rely on current equipment.

When two aircraft are within a 3.1 km band around the radar, their transponder responses overlap. Each transponder sequence is $21\mu$s long, and the delay between the interrogation transmission and the reception of the transponder response is equal to the time needed by the EM wave to travel from the radar to the aircraft and back [Stevens, 1998]. Therefore, the minimum separation to avoid data collisions is:
If we cannot separate these two responses, the difference between the distances from the ground sensor of the two aircraft will have to be greater than 3.1 km, and the flow of air traffic will be slowed down. Presently, the ATC controller guides the pilots to maintain the required separation, but to do so the two aircraft have first to be detected. If the pilots fly in visual conditions, they have to maintain the separation by themselves. When two transponders response collide, the received signal is longer than normal. Currently, the ground sensor just throws the whole sequence away, and waits for the next one. If only one sequence is missed, the air traffic system will not be perturbed (SSR usually work at an update rate of 2Hz). But if a large consecutive number of transponder responses is missing, the aircraft may disappear off the radar display. This scenario can be a problem since a data collision means two aircraft are in the same 3.1km band in space.

This chapter presents an attempt to separate two overlapped transponder responses. First of all, when the aircraft response length is above the normal length, the SSR receiver can detect the existence of two overlapped signals. When the two signals are not synchronized (if they are not received exactly at the same time), the SSR can detect the front end of the response as one signal, and the back end as another. Another solution consists in demodulating the overall signal and attempting to separate the two signals with a logic circuit. These two approaches achieve a certain degree of success. This chapter presents a third possibility.

As electronic systems become more and more sophisticated, transponders produce more accurate and stable replies. Based on this stability, we will try to analyze overlapped replies by their pulse shape, and then we will see how the system performance is affected. To do so, we need to look at the sequence parameters - such as pulse width, exact frequency or phase - and to determine them for each received sequence. The frequency and the phase of the received signal are main factors and can greatly help in separating each sequence. To
determine the frequency and the phase of the signal accurately, we will use a Phase Lock Loop (PLL).

After selecting the sequence’s parameters and determining them accurately, we need a specific algorithm to recover the signals. We will use a shape comparison to determine a solution, and compare the two solutions - for the two overlapped sequences - to the original received signal. If it does not match, we will introduce a loop to determine new solutions and to compare them for a second time to the original received signal.

### 3.1 Description of Mode C transmission

As it is described in Michel C. Stevens’ book *Secondary Surveillance Radar* [Stevens, 1998], a received transponder reply pulse shape is more complex than a simple rectangle (see Figure 3.1).

![A trapezoidal transponder pulse. Typical shape of demodulated pulses at the output of a SSR receiver](image)

**Figure 3-1** A trapezoidal transponder pulse. Typical shape of demodulated pulses at the output of a SSR receiver

Each pulse can be characterized by four variables: the rise time, $t_r$ (defined as the time between 10% and 90% of the pulse final value), the pulse width, $t_w$,
(the time between 50% of the pulse final value on the front end and 50% on the back end), the fall time $t_f$, (defined as the time between 90% of the pulse final value on the back end and 10% on the back end) and its amplitude, $a$. Each transponder has its own set of parameters, within the range allowed by the FAA. Moreover, the distance between the aircraft and the ground-based antenna affects the amplitude of the received signal. Therefore, each one of these parameters can be seen as a feature of the reply from a particular aircraft.

The pulse width used in Mode C (both interrogation and response signals) has been fixed at $0.45 \pm 0.1 \mu s$. The rise time is typically equal to $0.05 \mu s$, and the maximum fall time is $0.2 \mu s$. In our simulation, we will use a typical pulse width of $0.5 \mu s$ and a rise time and a fall time both equal to $0.05 \mu s$. These are very strong approximations and we will need to demonstrate later on that our system also works with all FAA signal specifications.

But first of all, a transponder never replies at exactly the center frequency of 1090MHz. Figure 3.2 describes the frequency distribution. Most transponders reply at a frequency of 1090 MHz $\pm 5$ MHz.
So our first task will be to detect the signal’s frequency, as well as its phase and then identify each signal's bit as a logical one or a zero. The radar receiver's demodulator needs to know the frequency of the signal and how to rebuild the signal. Moreover, the frequency and the phase are two characteristics of a sequence and vary from one transponder to another (Figure 3.2). Therefore, by
accurately recognizing the phase and the frequency of the received signals, one can separate the two overlapped signals. One way to do so is to use a Phase Lock Loop (PLL). The PLL locks at the input signal’s phase only when the Voltage Controlled Oscillator (VCO) and the input signal have the same frequency. Therefore, when the PLL is locked, one can detect the phase and frequency of the input signal.

### 3.2 Phase Lock Loops (PLL)

#### 3.2.1 PLL Basics

The received signal from an aircraft transponder will have a random frequency within a certain band (see Figure 3.2), and a random phase. A Phase Lock Loop (PLL) is used to lock onto the transponder signal’s phase and frequency. A PLL is a feedback system in which the feedback signal is used to lock the output frequency and phase to the phase and frequency of the input signal [Smith, 1997]. A commonly used architecture is shown in Figure 3.3.

![PLL block diagram](image)

**Figure 3-3** PLL block diagram (Kd is the multiplier gain, Kv is the VCO gain, Va is the multiplier output voltage, Ve is the error function at the filter output, and V0 is the Voltage Controlled Oscillator output voltage)

The input signal is the transponder signal, received by the ground sensor, and down converted to an intermediate frequency. The error function is equal to
the difference between input signal phase and the detected phase. The VCO output is a waveform whose phase is proportional to the error function, with a multiplicative factor equal to $K_v$. When the difference between the VCO output frequency and the PLL input frequency is within the PLL lock range (see Appendix B), the error voltage $V_e$ drives the VCO frequency toward the PLL input frequency.

When the two signals have the same frequency, the loop locks with the VCO output phase $\phi_2$ at 90° to the input signal $\phi_1$, as shown on Figure 3.4 (see Appendix B for further theory on PLL). When the difference between the received signal phase and the VCO output phase is equal to 90°, the multiplier output is a DC signal. Then, the VCO command signal is a DC signal, therefore the VCO output keeps the same phase, and the loop locks.

![Figure 3-4](image)

**Figure 3-4** Path along the phase detector characteristic. When the difference between the VCO output phase and the PLL input phase is $\pi/2$, the PLL locks.

When $0 < \phi_1 - \phi_2 < \pi$ and the VCO input is positive ($\cos(\phi_1 - \phi_2) > 0$), the VCO output phase increases and the loop leads the phase difference toward $\pi/2$. The opposite happens if the VCO input is negative ($\cos(\phi_1 - \phi_2) < 0$): the output of the LPF is a negative voltage and the VCO phase decreases toward $\pi/2$. But if the input function changes too quickly, the error function will jump from one positive value to another negative value and the PLL will never lock on. VCO works similarly for $-\pi < \phi_1 - \phi_2 < 0$, as shown on Figure 3.4. For more specific explanation about PLLs, the reader should refer to Appendix B. Our task is first to
simulate a PLL, and to use this PLL to recover two overlapped ASK signals (code given in Appendix A). Initially, we will look at a use of PLL without time constraint and without any interfering signal or noise. Then, we will see the effect of disturbances that may occur in practice.

### 3.2.2 Normal Use

We are going to simulate a PLL with a first order Butterworth LPF in order to study the influence of the overlapping over the phase detection, as well as some parameters (noise, frequency mismatch, etc). The PLL will also be useful to detect the signal frequency. Our first input signal will be a single sinusoid. After that, we will introduce various elements (noise, overlapping signal, frequency mismatch, etc) describing what may happen in real cases. The simulation of the PLL was run for a set of values corresponding to a critical dumping factor ($\xi = 0.707$). Keeping Figure 3.3 notations:

* A signal frequency of 30 MHz
* $K_d = 5$ V/rad
* $K_v = 1/10$ MHz/V
  
  * A first order Butterworth LPF, cut-off frequency 1MHz, -6dB/octave

All the simulations of this chapter have been run for this set of parameters, except when specified differently. The VCO phase output is shown on Figure 3.5.
Figure 3-5  VCO output phase for a single sinusoid at the input and a critical damping factor

The response of the loop is quite fast (stable at 0.3 µs), and corresponds to a typical second order filter’s response. We get an overshoot of \((0.5 - 0.45)/0.45 = 12\%\) and an undershoot of a few percent. These results correspond to the second order PLL linear transfer function (see Appendix B for further details):

\[
H(f) = \frac{\omega_o^2}{\omega_o^2 + 2\xi \omega_o f + f^2} \tag{Eq. 3-2}
\]

where

\[
\omega_o = K_d \times K_v \times \omega_L \quad \omega_L \text{ is the LPF cut-off frequency} \tag{Eq. 3-3}
\]

\[
2\xi = \left\{ \frac{\omega_L}{(K_d \times K_v)} \right\}^{\frac{1}{2}} \tag{Eq. 3-4}
\]

The PLL input signals are composed of a set of 15 ASK bits, and necessarily begin and end with bits equal to one, called P1 and P2. The recovery of overlapped signals relies on the P1 pulse to detect the beginning of the first received signal, and to compute the frequency and the phase of this signal. If two signals are completely overlapped, that is, they both start and end within 0.5 µs, we cannot separate them and must throw the signal away and wait for another
signal to arrive. Exact alignment of two signals rarely occurs (the two aircraft have to be at the exact same distance from the radar receiver with a 30 meters tolerance due to pulse width), and relative motion of the two aircraft will cause the signals to separate in time at the next radar interrogation.

### 3.2.3 PLL VCO at the Wrong Frequency

Here is what happens if the free running frequency of the VCO is not equal to the input signal’s frequency. Figure 3.6 describes the error voltage $v_e$ with respect to time for a 10% frequency difference between the initial VCO frequency and the PLL input signal frequency (see Appendix B for further details on the PLL theory).

![Figure 3-6](image)

**Figure 3-6** PLL output for a single sinusoid at the input. VCO at the wrong frequency (VCO initial frequency 33Mhz is 10% higher than the received signal frequency equal to 30MHz)
The PLL does not lock on, and its output is just the difference frequency between the VCO and input signal frequencies. Let us see how this can be explained mathematically (for more developed mathematics, see Appendix B):

Let $f_1$ and $\phi_1$ be the frequency and the phase of the input signal, and let us keep the notation of Figure 3.3. The VCO output and the PLL input amplitudes are normalized. This can be done in a SSR receiver detecting the received signal amplitude and using a commanded linear amplifier. The output of the multiplier is $v_0$, where

$$v_0(t) = \sin(2\pi f_1 t + \phi_1) \times \cos(2\pi f_2 t + \phi_2) \quad \text{(Eq. 3-5)}$$

If $(f_1 - f_2) \neq 0$, then a difference frequency appears at the output of the multiplier, as seen Figure 3.6, where $f_1 = 30\text{MHz}$ and $f_2 = 33\text{MHz}$.

Therefore, the first problem with our system is to detect the input signal frequency. One solution consists in trying to lock the PLL for different frequencies. When we obtain the right output format (the VCO output phase does not change any more after a certain amount of time), we know that $(f_1 - f_2) = 0$ and that we locked on the input signal’s frequency. At that point, the PLL output will be stable, and can therefore lock to the input signal’s phase.

### 3.2.4 Noisy Environment

Now, all the simulations we have run so far were done for a noise power equal to zero. Let us look at the effect of channel noise on phase detection. Figure 3.7 shows the result for a Signal-to-Noise (S/N) ratio equal to 10dB in the equivalent noise bandwidth defined further on. The S/N ratio is generally at this level for SSR with an aircraft at maximum range (between 10dB and 15dB). As we can see, we cannot wait for a stable output; otherwise, we will never lock on. A certain error margin has to be introduced, and we will see later that it does not have any effect on the overall system’s result. Since the channel is AWGN, averaging the PLL’s output, we will come up with phase close to the input
signal’s phase. For a small phase error, the PLL can be modeled by a linear system where the phase detector is replaced by the subtraction of the PLL input signal and the VCO output signal. Therefore, AWGN noise at the PLL input generates AWGN noise at its output. To the channel noise, one has to add jitter noise introduced by the sampling process. The equivalent noise bandwidth $B_n$ is equal to [Blanchard, 1976]:

For a first-order loop:

$$B_n = K = K_v \times K_d$$  \hspace{1cm} (Eq. 3.6)

For a second-order loop:

$$B_n = \frac{\omega_0}{2\xi}$$  \hspace{1cm} (Eq. 3.7)

This simulation uses a second-order loop. Then the noise equivalent bandwidth is equal to 354 kHz.

![Figure 3-7](image.png) PLL output in a noisy environment  \hspace{0.5cm} S/N = 10dB for a noise equivalent bandwidth of 353 kHz
3.2.5 Two Signals Added

When two aircraft are present within a 3.1 km range band, as measured from a SSR, their radar responses overlap in time. Consequently, the radar receiver is unable to process the information and the two aircraft may disappear off the Air Traffic Control radar display.

Therefore, the case that really interests us in this paper is the case where two ASK signals overlap each other. What will be the result for two logical one bits overlapping (i.e. for two sinusoids)? We are not concerned in the other cases: a logical one overlapping a logical zero, or two overlapped zeros. Let us assume that the PLL has already locked onto the first sinusoid, and that another one arrives, as shown on Figure 3.8.

![Figure 3-8](image)

**Figure 3-8** One signal overlaps another in time

In Figure 3.9 we see the response for the first signal (similar to Figure 3.5), followed by the same response for the second signal 2 µs later. Therefore, the second signal does not destroy the system’s stability; the PLL locks on 0.5µs
after the beginning of the second signal. Actually, the sum of two sinusoids is just another sinusoid, whose phase is equal to half the sum of the two sinusoids’ phase (assuming equal amplitude).

Figure 3-9  PLL output for two overlapping signals at the same frequency, first signal starting at $t = 0$, second signal starting at $t = 2\text{ms}$

However, this case never occurs, since the two signals’ frequencies would have to be exactly equal. Two different transponders are very unlikely to have the exact same frequency. The worrying case is also the most common: two overlapping signals coming from two different transponders, with a significant difference in frequency. Figure 3.10 shows the PLL output for two overlapped input signals, with a frequency difference of 10% (first signal at 30MHz, second signal at 33MHz. First signal starts at $t = 0$ and the second signal starts at $t = 2\text{\mu s}$). The PLL locks on the first phase (similarly to Figure 3.5), but when the
second signal is introduced, the PLL cannot lock on any more. Let us review the expression for the sum $V$ of two sinusoids, with $v_1$ and $v_2$ having different phase $\phi$ and frequency $f$:

$$v_1(t) = \sin(2\pi f_1 t + \phi_1) \quad \text{(Eq. 3-8)}$$

$$v_2(t) = \sin(2\pi f_2 t + \phi_2) \quad \text{(Eq. 3-9)}$$

$$V = 2 \sin(\pi (f_2+f_1)t + \frac{1}{2}(\phi_1+\phi_2)) \times \cos(\pi (f_2-f_1)t + \frac{1}{2}(\phi_1-\phi_2)) \quad \text{(Eq. 3-8)}$$

Therefore, when the two frequencies $f_2$ and $f_1$ are equal, the cosine term is equal to 1, and the only term remaining is the sinusoid, whose phase $\Delta \phi$ is

$$\Delta \phi = \frac{1}{2}(\phi_1+\phi_2) \quad \text{(Eq. 3-9)}$$

the mean of the two phases. Moreover, as the PLL needs only 0.5 $\mu$s to lock on, for a small difference of frequency, the result can still be kept. But, when the two frequencies $f_2$ and $f_1$ are different, the cosine term begins to modulate the amplitude of the sine term, and the phase of the first signal is lost, as seen in the Figure 3.10
Therefore, with this system, we can detect the transponder’s response phase, and also its frequency. We just have to look at the PLL error function, and as long as we detect an oscillating error signal (similar to Figure 3.6), we know that the chosen frequency is not equal to the transponder’s response frequency. When we scan to a different frequency and observe at a certain point a stable PLL output, it means we have detected the transponder’s frequency.

Let us note that to do so, we will have to record the received sequence. This means our system will have to sample the transponder response first, before using a digital-PLL [Rohde, 1983].

**Figure 3-10** PLL output for two overlapping sinusoids with different frequencies
(First Signal with a frequency of 30MHz, starting at t=0, second signal with a frequency of 33MHz, starting at t=2ms)
3.3 Demodulation problem

Air Traffic Control (ATC) relies on Second Surveillance Radar (SSR) to identify the aircraft by a squawk code and locate it in azimuth range and altitude. The SSR interrogates the on-board transponder, and receives the transponder ASK response composed of 16 bits (see Appendix B for further details about transponders formats). When two aircraft are too close to each other (within the same 3.1 km range band), their transponder responses collide. In that case, demodulation requires separation of the two signals.

We will now use the PLL in the demodulation process with two overlapping signals. First, we will try a classical demodulation, and then we will try to demodulate the signal using the shape of received bits. Finally, to improve the system’s performance, we will use a loop to correct the eventual demodulation errors.

3.3.1 Simulation for overlapped signals

The garbling effect occurs when two aircraft respond to the same radar interrogation. When this phenomenon occurs, the secondary radar cannot separate the two responses. Let us focus our work on the de-garbling problem. The solution often used to de-garble is to detect when two replies have collided and to throw the whole signal away. We can obtain better results, and at least keep the first reply.

After reception, the signal is sampled and store in a memory. This memory gives time to the PLL to detect the input signal’s frequency and phase. The demodulation process consists in detecting the frequency and the phase of two transponders responses with a PLL, before multiplying the received signal by the PLL VCO frequency, once the PLL is locked on the first signal. The following block diagram shows the system we are using to demodulate the ASK sequence.
Figure 3-11  Block diagram of the demodulator for a single (not overlapped) transponder response. After frequency and phase detection, the received signal is multiplied with a sine wave corresponding to the same frequency and phase, before being filtered.

When two signals are overlapped, the separation is quite similar to the single demodulation described above. Once the second sequence is filtered out, and once the first sequence is rebuilt, it is subtracted from the received signal. The output of the subtraction is equal to the second response. This response is then demodulated (Figure 3.12).
Figure 3-12  Block diagram of the demodulator, separating two overlapped transponder responses. Once the first sequence is demodulated, it is subtracted from the received signal. The subtraction result is then demodulated the same way, and the second sequence is obtained.

Let us use a mode A transponder reply (see Appendix B for details), with a first aircraft squawk code 4321 (A=4 B=3 C=2 D=1, overall bits F1 0 0 1 0 0 1 0 1 1 0 0 0 F2 SPI), and a second aircraft with squawk code 4322 (A=4 B=3 C=2 D=2, overall bits F1 0 0 1 0 0 1 0 1 0 1 0 0 F2 SPI). Typically, these aircraft can be small planes, which have taken off from the same airport, and therefore have similar squawk codes. Let us suppose they are both on approach to an airport equipped with a SSR or a ground based TCAS unit, with half a kilometer between aircraft. Then, they are replying to the same interrogation, and the reply from the second aircraft overlaps the first by 1.5µs. On Figure 3.13, we can see what the received signal looks like when the two transponders’ frequencies differ by 20kHz.
Figure 3-13  IF received signal for two replies overlapped by 1.5ms (frequency difference of 20kHz, S/N = 15 dB)

From this signal, we can perfectly extract the two signals, and recover them in most cases. Here are the outputs of the demodulators, on Figure 3.13 and 3.14. There are no bit errors out of this process, because the two signals respect certain parameters. When their frequencies become too close to each other, or when there is a large difference in their amplitudes, demodulation is more difficult.
Figure 3-14 Recovery of the first signal, for two overlapped signals with different frequencies (f1=30MHz, f2=33MHz)

On the top graph of Figure 3.14, we can see the first signal envelope. The second signal envelope looks like AM. Actually, its amplitude is modulated by a factor $\alpha$

$$\alpha = \sin(\pi (f_2-f_1) t + 1/2(\phi_1-\phi_2)) \)  \text{ (Eq. 3-10)}$$

Therefore, if the two frequencies are separated sufficiently, the system will neglect the second signal. That is what is happening above; the system did not make any mistake.
Figure 3-15 Recovery of the second signal for two overlapped signals 
(f1=30MHz, f2=33MHz)

Now, we can study the effect of different parameters on this process, and particularly how much the two frequencies have to be different to permit separation of the signals, and the minimum separation time between the pulses.

Figure 3.16 gives us the bit errors in the recovery of the two signals with respect to the overlapping time. Despite the system can correctly recover the first signal up to six bits overlapped, the second results are much more random, beginning at the very first bit.
Therefore, the system cannot separate the two aircraft for an important number of bits overlapped. But the system has been able to recover the first signal in particular conditions (no more than 5 bits overlapped). So far, the system separates the two signals only by their frequency. Other parameters are available and specific to each transponder: the transponder’s response angle of arrival, the phase, the pulse width and the amplitude.

3.3.2 Use of Monopulse

*Monopulse* is a technique for the accurate measurement of the angle of arrival of the signal. The antenna beam is divided into two beams, called Sum and Difference beams (Figure 3.17). The difference beam gain has a deep null at 0°. The sum beam has its highest gain at 0°, and its 3dB angle is only 3°.
When a radar sensor using monopulse detects an aircraft response, the system subtracts the power received on the difference beam from the power received on the sum beam. If the result of this subtraction is negative (the power received on the difference beam is greater than the power received on the sum beam), it means the aircraft signal angle of arrival is larger than 3°. Moreover, if two responses are overlapped, monopulse can separate them; but the solutions provided are only given with a certain probability. It is impossible for the system to tell if a solution is right or wrong.

Overlapping transponder response occurs when two or more aircraft are within a 3.1 km distance as seen by the radar. One way to separate these aircraft radar response is to detect their angle with respect to the receiver. The monopulse technique uses two beams: a sum beam (narrow, around 5 degrees) and a difference beam (broader with a deep null at zero degrees). Since the two aircraft A1 and A2 are detected at different angles, A1 will give a stronger response for one of the beams, and A2 for the other. With this difference in amplitude, we are able to distinguish them if the bits do not overlap (no logical ones overlapped). We can know whose pulse it is simply looking at its amplitude.
3.3.3 Use of comparison

3.3.3 a) Generalities

Because of the improvements of the design and construction of transponders, the bit waveforms are very similar one to another for a single signal. We can use this similarity to link each bit with each signal, and therefore discriminate between two overlapping signals.

The main difficulty with this approach was to determine the bit parameters (rise time, pulse width, frequency, phase), with sufficiently good accuracy to detect two different signals. The PLL is greatly useful here, detecting the frequency and the phase with accuracy.

In this way, we can demodulate the first sequence, before rebuilding the sequence with the frequency and the phase detected, and then subtract it from the received signal to recover the second sequence. This subtraction is only possible because of the accurate detection of all the pulse’s parameters. Using this method, we obtain a correct demodulation for all overlapping cases as soon as the frequency difference is greater than 90kHz, as one can see in Figure 3.18.
The frequency difference used for this simulation is 90kHz (f1=30MHz and f2=30.9MHz). The two graphs on the top describe the performance of the first method (number of errors with respect to the number of bits overlapped). The two graphs on the bottom describe the performance of the new method. The first method separates the two signals based upon the frequency (see Figure 3.12). For a small frequency difference, the first method is not sufficient: on Figure 3.18, both signals are erroneous at 10 bits overlapped (overlapped time: 5µs). On the other hand, the second method does not make any mistake for any overlapping time.
3.3.3 b) Impact of frequency difference on results

For smaller frequency differences, errors occur, as shown on the Figure 3.19, for a frequency difference of 60kHz.

![Figure 3-19](image)

**Figure 3-19** Relative performance of the two demodulation methods for a frequency difference of 60kHz (number of errors with respect to the number of bits overlapped)

These results do not satisfy us, since an error occurs in the second method demodulation for the first signal (left bottom figure), for 24 bits overlapped (overlapping time: 12µs). There is a high probability that two different transponders frequencies are within 90KHz (refer to Figure 3.2). Therefore, we need to improve our demodulation to take that into account.
After the first demodulation, we can reconstruct the first and the second sequence separately, because of the accurate detection of the pulses’ parameters. Therefore, running again the demodulation for these new reconstructed signals, we have a greater chance to obtain the right sequences. We can go through this process as long as the sum of the reconstructed sequences does not match with the received signal. For instance, a SSR sensor receives the following sequence: 1001100100100010100101. The first demodulation gives the following sequences: 1001100110001 and 1000110100101. The overlapping time is the difference between twice the normal sequence length and the actual length of the received signal: 2*12 – 22 = 6 bits. Therefore six bits are overlapped. If the demodulation were correct, the sum of the two signals would be: 1001100100100010100101. The received 11th bit is a logical one, whereas the sum of the two signals’ 11th bit is a logical zero (in bold in the sequences). Therefore, either the 11th bit of the first signal or the 2nd bit of the second signal is actually a logical one. The system will rebuild the two signals, sum them and compare the sum with the received signal in these two scenarios.

With such a loop, here is what happens for two overlapping signals with a frequency difference of 20kHz. On the following graph, one can compare the performance of the loop. The graphs on the top describe the results for the first demodulations, and the ones on the bottom row the results after the second demodulation.
Figure 3-20  Two consecutives demodulations for a frequency difference of 20kHz

After only a couple of consecutive demodulations, we can detect the right sequences for only a 20kHz frequency difference. Now, let us see the impact of mismatches in rise times, detected frequencies, and detected phases.

3.3.3 c) Impact of rise times on the results

Theoretical rise time of transmitted pulses is 0.05$\mu$s for all transponders. This figure has little impact on results: even with a 10% difference, the results concerning the demodulation errors stay the same.

Introducing the loop in the demodulation presented in section 3.3.3.b), the performance is obviously greater. Instead of demodulating the signal once, we can take advantage of one demodulation for the second one. Let us imagine that we came up with a demodulated signal $S_1$ for the first sequence and $S_2$ for the
second one. Since we know the frequencies and phases for each one of the sequences, we can rebuild the incoming signal from S1 and S2, if the results do not match with the received signal, we can detect a demodulation error, eventually locate it, and try a second time (see previous section).

The transition time is defined as the time between which the pulse amplitude goes from 10% of its final value to 90% ($t_r$ in Figure 3.1). The falling time $t_f$ is the equivalent of the rising time for the back end. We will assume that the rising time and the falling time are equal, and simply call them transition time. On Figure 3.21, one can see what happens for a transition time of 0.06µs.

![Figure 3-21](image)

**Figure 3-21** two consecutive demodulations for a transition time of 0.06ms (a 60kHz frequency difference)
When we increase the rise time, we add another problem to the demodulation problem. It is more and more difficult for the PLL to lock on, since of the delay period while the signal is not a regular sine wave. Moreover, for higher transition times, a logical one is more difficult to detect since the pulse width is limited.

With a rise time such as 0.10µs, errors occur in the second loop (Figure 3.22). As one may see in the Figure, the results after the second demodulation are inexact for high overlapping times.

**Figure 3-22** two consecutives demodulations for a transition time of 0.10ms

Therefore, we need to improve our performance in transforming the PLL. If the rise time becomes too important (i.e. when the envelope detector reveals that the time difference between 10% and 90% of the final amplitude value is too
important), it is possible not to start the PLL in the very beginning of the pulse, but a little bit later.

On Figure 3.23, one can see the PLL’s output for an input pulse with 0.45μs rising time, and a PLL input that is not delayed. The PLL needs around 2μs to lock on correctly. This is obviously a problem when the pulse lasts only 0.5μs. A solution consists in storing the signal, and repeating the 0.5μs sequence. But this process supposes perfect synchronization in order to avoid unwanted phase shifts. Another solution is presented below.

![Figure 3-23 PLL output for a rise time of 0.45ms, non-delayed input](image)

Figure 3.24 shows the same input signal with the same rise time, but the PLL’s input has been delayed. Instead of feeding the PLL at the very beginning of the received signal, we wait 0.6μs. In this way, the transition period does not seriously affect the PLL output. Therefore the PLL locks on within the pulse length: 0.5μs.
3.3.3 d) Impact of mismatch in the detected phase

So far, we assumed that the PLL was working correctly and that the detected phases were the actual signal phases. Now, if we simulate an artificial error of 30% in the detected phase, correct results can be reached fairly quickly (i.e. with only two consecutive demodulations). Instead of using the PLL result, the simulation simply uses 1.3 times the phase used to generate the received signal. The results are presented on Figure 3.25. The first signal does not present any error, and the second signal has only one error, for 21 overlapped bits.
3.3.3 e) Impact of the signal amplitude

So far, results have been obtained for two signals with equal amplitudes. For signal with different amplitudes, the results can be very different from one to another. Let us see what happens for a 3dB difference in amplitude. The results are presented on Figure 3.26.
Figure 3-26  3dB amplitude difference between the two signals (amplitude one for signal 1 and \( \frac{1}{2} \) for signal 2, with a 60kHz frequency difference). No error detected.

Now for 10dB difference in the signal amplitude, let us see the results on Figure 3.27:
The main problem we can observe is the difficulty to detect the amplitudes accurately, because of the sampling effects of the received signal at the ground sensor. So we can assume that before sampling, we detect the correct amplitudes in analog form. Let us notice SSR receivers use AGC to find the average signal level, and filter out the signal components that are below a threshold.

Nevertheless, let us note that for two transponders responses to be overlapped, the aircraft cannot be separated by more than 3.1km. Therefore, for two overlapped signals, the responses amplitudes are always similar. When both the aircraft are at least 10km away from the radar sensor, the amplitude difference is bounded by 5dB.

Figure 3-27  10dB difference between the two signals amplitude (signal one amplitude equal to 1, signal two amplitude equal to 0.1, with a 60kHz frequency difference). Error occurs for the second (the weakest amplitude) signal
The path loss $L_p$ is equal to:

$$L_p = 20 \log_{10} \left( \frac{4 \pi R}{\lambda} \right) \quad (\text{Eq. 3-11})$$

Where $R$ is the distance and $\lambda$ is the wavelength of the signal. The amplitude difference for 10km is $\Delta L_p$

$$\Delta L_p = 20 \log_{10} \left( \frac{10 \text{km}}{13.1 \text{km}} \right) = -5.4 \text{ dB} \quad (\text{Eq. 3-12})$$

In brief, we can conclude that for a high difference in amplitude, there is a chance for the demodulator to fail.

### 3.4 Performance of this system

We have shown that to recover two overlapped signals is possible for nonsynchronized overlapped transponder responses. Now, with the simulator developed by Eric Shea [Shea, 2002], we are going to show how this affects the performance for small airports, in terms of delays between landings and departures. The simulation shows the performance of a Smart Landing Facility system based on the TCAS technology. A ground sensor, called “TCAS on a stick” interrogates TCAS equipped aircraft. TCAS works like Secondary Surveillance Radar. An on-board TCAS sends interrogations to another on-board TCAS. The latest TCAS responds to the first by broadcasting the aircraft squawk code and altitude. “TCAS on a stick” keeps the same organization, except the interrogator is on the ground.

Eric Shea has simulated in Matlab the number of data collisions occurring when a fixed number of aircraft arrive during an hour, from random directions, land or take off. These aircraft arrive randomly in the airspace, go directly to an arrival point, then turn toward the runway and follow a glide slope of 3 degree to the runway surface. Similarly, the same number of aircraft takes off randomly during an hour, follows a steady slope of 4.5 degrees until a departure point where they turn to a random direction. The ground sensor used in this simulation is a TCAS, and the on-board equipment is a mode C/S transponder. Using this
simulation, for a four sectors antenna, we can find the number of collisions with respect to the number of aircraft present in the airspace during an hour. Shea’s results [Shea, 2002] show that, even with the best system of antennas, 20 aircraft lead to 30% of data collisions, and a maximum outage time of 450 s (an aircraft disappeared off the ATC display for 450 seconds or 7 minutes 30 seconds). Of course this simulation does not take any procedure into account; the different aircraft arrive at random times and from random directions.

When we introduce the pulse shape analysis system, even with 60 aircraft arriving in an hour, we still do not have any triple collision (three overlapping responses). Therefore, we do not have any demodulation error at all. We could go further in the number of arrivals and departures, but it would not have any meaning, since 60 aircraft an hour means already one aircraft landing every minute. Moreover, when we introduce a two minutes separation between two departures or two arrivals, the system does not detect any data collision at all. This result will be discussed in the next chapter.

3.5 Conclusion

When two aircraft are within a 3.1 km range band around a ground Secondary Radar, their radar responses overlap. A solution to recover each response consists in collecting parameters accurately (frequencies, pulses widths and phases), and analyzing the shape of each received bit to determine whose it is. The PLL simulation has shown the influence of diverse parameters (mismatch in frequency, noise, amplitude difference, rise time, etc) on the parameters detection. When the system comes up with a potential solution, it can rebuild the sine wave and compare it to the received sequence. If the rebuilt signal does not match with the received sequence, the bits leading to an error are changed and the new signal is rebuilt and tested with the received sequence. This simulation can be used for further research on the influence of some other parameters over aircraft detection, or the system implementation.
This method can greatly reduce the delays between two aircraft landing or in departure. It requires only new ground equipment, and relies entirely upon the transponders. Nevertheless, the Air Traffic Control scheme may not be changed only for this system. This system may be used for emergency procedures: to track aircraft while they correct their flight to maintain the required separation.

The main limitation of this technique is the case of very different transponders responses amplitude. To determine accurately what is the critical amplitude difference, and if such a difference can occur with today’s FAA requirements over the transponders manufacturers, one needs to develop and test a prototype.

In the next chapter, we will see how we can adapt different approach procedures to take the GPS and TCAS technologies into account.
Chapter 4 – Minimal separation for GPS - TCAS equipped aircraft

Air Traffic Control (ATC) relies on Surveillance Secondary Radar (SSR) to identify and locate incoming or departing aircraft. SSR sends a signal called interrogation to an on-board transponder. This transponder answers back, emitting the identity and the location of the aircraft. But if two aircraft are within a 3.1Km band from the radar, their transponder responses overlap. As a consequence, they disappear off the radar display. That is the reason why ATC tries to maintain a five minutes separation between two arriving aircraft. This separation delays the arrivals. As the air traffic increases, ATC sometimes needs to increase this delay to maintain passenger safety.

Air traffic in the US is organized around 30 major airports, called hubs. However, the Small Aircraft Transportation System (SATS) wants to take advantage of the 5000 small airports to lighten air traffic around these hubs. Direct flights between small airports are not currently possible in all weather conditions. The SATS program is an attempt to develop a Smart Landing Facility (SLF) that small airports could buy and that would allow direct flights from a small airport to another small airport in all weather conditions. To guarantee equal performance with big airport in terms of delays and safety, this device needs work for a five minutes separation between two aircraft.

Using a digital system, and analyzing its received signal, a SSR system can separate two transponder replies that have collided. This process allows the aircraft to fly with a shorter safety distance and to be continuously detected by the ground sensor; therefore air travel is safer and more efficient. However, Secondary Radar is not a feasible solution for small airports because of its high cost.

The use of GPS and Traffic Collision Avoidance Systems (TCAS) allows greater freedom for the pilot, and improved opportunities to Air Traffic Control
(ATC). Originally, TCAS is an onboard piece of equipment. We propose to use this technology on the ground for SLF. This new equipment, called “TCAS on a stick”, would play the role of a SSR for small airports: to detect aircraft in the airport’s airspace, and to broadcast this information to all equipped pilots. Moreover, current ATC in small airports relies on visual detection from the pilot. A step into ATC automation consists in assisting both the pilot and the controller with current technology.

We now need to examine how to take advantage of these new technologies to avoid traffic conflicts and to increase the number of movements (aircraft landing or taking off) in small airports. We will also need to compute the limitations of the new procedures, in terms of number of movements allowed in an hour and conflicts that may not be solved. Finally, we will confirm our proposals with a simulation.

### 4.1 Goals of Smart Landing Facility

TCAS used at a SLF is called “TCAS-on-a-stick”. It uses Mode C transponder technology, and works like SSR, using an interrogator/responder device. The response is a 15 bits ASK signal of 20.8 $\mu$s. In Mode C, the first bit, as well as the last one, is always coded as a logical one in order to identify the signal. This 15 bits sequence contains the aircraft’s ID or altitude; the first response is the ID number, called squawk code, and the second the altitude, alternating between the two. Moreover, detecting the response angle of arrival and the delay between the interrogation’s emission and the response’s reception, the system can compute the location of the aircraft. TCAS-on-a-stick has to keep the same parameters, and work at the same frequency. This way, when an aircraft is interrogated by the SLF, the transponder will not see any difference between TCAS-on-a-stick and SSR.

The goal of this evaluation is to determine how TCAS-on-a-stick will affect current airport services, how many aircraft are allowed to take off or land per
hour and identify the limitations of this system. For TCAS equipped aircraft, the pilot may have enough information to fly a free flight. But the procedure needs to take into account aircraft without TCAS on-board.

4.2 Services Provided by TCAS at a SLF

TCAS-on-a-stick has to provide the same services as SSR. The most important point is to keep track of all aircraft at all times. Moreover, for better transparency, the new system has to work like the former one, respecting the current procedures and avoiding new equipment.

Nevertheless, the new system has to provide a better service for the ATC: to reduce the delays between aircraft and to keep track of the airplanes. This system should also help pilots flying free flight to detect and avoid conflict.

When two aircraft are located within a band of 3.1 km from the ground-based TCAS, or if the difference between their transponder responses angle of arrival is below the ground antenna beam-width (six degrees is the figure used for TCAS III), their transponder signals collide. In certain conditions, it is possible to separate them (see Chapter 3); however, when the number of aircraft becomes too large, one cannot separate them.

In small airports, pilots repeatedly practice landing and taking off. To do so, they have to turn around the airport, often at a distance below 3.1 Km from the ground station. Their transponder responses are therefore likely to collide. Nevertheless, we have seen in the previous chapter (Chapter 3) that one may be able to demodulate two overlapped signals correctly. For better safety, we will keep the optimum separation of a 3.1 Km safety distance and a six degrees angle from the ground sensor.

To be approved by the FAA, a transponder has to emit $0.5\mu s \pm 0.05\mu s$ wide pulses. Using this boundary, one can compute the minimum separation between two aircraft in order to keep their transponder responses unsynchronized. Let two transponder responses be overlapped: one transponder emits $0.55 \mu s$ long bits,
and the other one 0.45 µs long bits. Figure 4.1 shows an example of two bits coming from these two aircraft separated by only 30 meters. If one bit is entirely overlapped (synchronized responses), it may be difficult to recover it, since the PLL (see Chapter 3) cannot detect its phase or frequency.

![Figure 4-1](image)

**Figure 4-1** Two transponders pulses, coming from two aircraft 30m away from each other

In this chapter, we will assume that all aircraft are separated by at least 30 meters.

### 4.3 Smart Landing Facility parameters

For our evaluation, we have assumed the pilots have followed FAA regulations and procedures. Nevertheless, with GPS, pilots can locate themselves more accurately than visually. By locating his or her aircraft in the airspace, the pilot will have a safer and quicker approach to the airport (Figure 4.2). The usual approach path consists in following several steps in altitude, and a three-degree glide slope to the airport.
The steps are useful for the pilot in particular to check out his or her position, and to keep the pilot clear of all obstacles. But if the aircraft is SATS equipped, these steps are not required anymore. On the contrary, the aircraft could begin its approach much later, eventually with a steeper angle, or with greater liberty. Consequently, the approach area around the airport could be smaller, therefore reducing the risk of collisions, and the delay between two landings. Moreover, in mountainous areas, a later approach is safer since the pilot stays away from mountains. Reducing the approach area, the risk of data collision is also reduced since the aircraft are less likely to operate in this region.

We need to study the different configurations leading to data collision and to propose SLF parameters acceptable to the FAA and the pilots. The optimum solution should allow a maximum number of movements (departure or arrival), which is 20 per hour. The factors to study are therefore the number of movements per hour and the approach configurations.

90% of SSR have a range of 350Km [Stevens, 1988]. But TCAS-on-a-stick range would be around 40Km. So we can consider that any aircraft in departure or arrival can be seen by the SLF, as long as its transponder signal is not overlapped by another aircraft transponder.

For such an overlap to occur, the two aircraft have to be within a common 3.1km band (1.7nm), away from the radar. Moreover, the SLF antenna has to see the two aircraft at the same time: if the antenna has a \( \theta \) degree beam width, the two aircraft have to be separated by no more than \( \theta \) degree from the ground.
sensor. Now, we are going to compute \( \theta \) in order to comply with the initial requirement (20 movements per hour, or one every three minutes).

Let us apply these figures to the Blacksburg GPS approach. The IAF (Initial Approach Fix) point is a point in the airspace where all arriving aircraft have to go before beginning the approach procedure. This point is at 12nm (around 24km), 197 degrees from North relative to the airport. At 24km, a \( \theta \)-degree angle makes a \( L \) km arc where

\[
L = 24 \times \theta \times \frac{2\pi}{360} \tag{Eq. 4.1}
\]

If the two aircraft fly at 150 kilometers per hour (average speed for small aircraft around airport), a three minutes separation is equivalent to

\[
L = 150 \times \frac{3}{60} = 7.5 \text{ Km} \tag{Eq. 4.2}
\]

Therefore,

\[
\theta = 18^\circ \tag{Eq. 4.3}
\]

This beam width can be achieved with current TCAS systems. For safety, we will use a 6° antenna beam width (twice the minimum beam width for a TCAS III antenna). With these parameters, to avoid data collision, the aircraft are to stay separated by more than 6° from the ground sensor and keep a safety distance between them of 3.1 Km.

### 4.4 New maneuvers using SATS equipment

Instead of using a statistical approach, we will study the configurations leading to a conflict, and propose a solution for each one of them. As a single data conflict can lead to a traffic conflict, it is important to look carefully at each of them. Then, we will propose a solution, and check if it leads to a conflict. A solution is said to be satisfactory if it does not lead to any data conflict.

We know how close two aircraft must be in order to create a data collision. Today, approaches do not prevent data collisions. Therefore, we need to study which procedures should be changed and to propose new ones.
The FAA cannot change all the procedures and facilities at once. It has to be done in several steps. The first one consists in introducing new procedures, while keeping the current equipment. Moreover, the new procedures have to be close to the former ones, so that the pilots will still have familiar approaches. Due to eventual technological developments in the future, the new procedures should be close to the current GPS procedures (Figure 4.3).

The changes will first happen on the ground, with the introduction of new procedures for Smart Landing Facilities. To keep close to the current situation, we will start with en-route procedures from the en-route ATC system (a pilot willing to land follows a radar vector under ATC control until he or she can begin the landing approach). This will not change. What is going to change is the ATC sequencing of movements to avoid data collisions and to track each aircraft. We also have to introduce a procedure for aircraft maneuvering around the airport, to avoid a potential conflict when heavier traffic is present in the airspace.

Moreover, we need to assume all aircraft will not get equipped at the same time. Many aircraft will keep Mode C transponders whereas others will use the new SATS equipment. Therefore, the SLF needs to support both, identify the type of equipment of each aircraft and interrogating them according to the type of transponders they are equipped with.

But if restrictions are to be applied to free flight, who is going to apply them? For small airports, the current practice consists in letting VFR pilots select their own approach path and visually detecting by themselves potential conflicts. If the pilots are the ones to decide in case of a potential traffic conflict, the FAA has to give them a precise set of procedures, and the means to avoid radar data collision. In that case, the on-board equipment will have to be upgraded, with TCAS for example. If ATC are the ones to decide, they need to detect the data collision, and then to broadcast instructions to the pilots.
4.4.1 An aircraft leaves the airport and meets an arriving aircraft

Let us first study the case of an arriving aircraft following the correct procedures, and another aircraft crossing its path. What should be the safe separation in time and distance between them?

Each airport has its own particular set of procedures for approaches. Approach plates (Figure 4.3) help the pilots for each airport in flying around each airport. These plates describe headings, topography (mountains, rivers, etc), points in space used for maneuvers (Taber, Sunny and Zooms in Figure 4.3) and missed approach procedures.
Let us consider the case of an aircraft coming from Roanoke, and looking for the Blacksburg IAF (HAWTO in Figure 4.4). This aircraft has been asked by Roanoke ATC to follow a radar vector toward the IAF (294° on the plate). Now, another aircraft has taken off from the Blacksburg airport and is going toward the North West (ZOOMS on the plate). Their two paths have to cross (see Figure 4.4); Blacksburg ATC cannot ask the second pilot to turn around HAWTO by the
South because it takes too long. Therefore, the task of ATC is to determine the minimal separation between the two aircraft, and to give instructions to the pilots.

Figure 4-4 An aircraft leaves Blacksburg airport and crosses the path of another aircraft arriving from Roanoke

The current practice for Visual Meteorological Conditions consists in listening to Blacksburg airport voice radio on CTAF, 123.05 MHz, and detecting visually other aircraft. Now, if the SLF provides to ATC the time and location of data collision, ATC will be able to warn these two aircraft and correct their path. When a data collision is detected, there is a risk of physical collisions. The physical collision can be avoided simply by assigning two different altitudes to the aircraft. But transponder response collisions remain. To avoid data collision, or at least to minimize the data collision time, ATC needs to control the separation of the two aircraft: when one of the aircraft is crossing the approach path, the other one has to be at least 3.1Km away, and the two aircraft and the ground sensor have to form an angle of at least six degrees. We know (Equation 4.2) the minimum separation between two aircraft flying 150km/h is five minutes. This is true when the two aircraft paths are parallel. When they cross, the separation time depends on the angle formed by the two paths.

ATC can introduce a delay between two aircraft in three ways. The first one consists in postponing the departure of the second aircraft (this is a current
practice, but it takes a long time). The second is to change the two aircraft paths, by changing their heading angles. This is possible when both pilots can detect each other, with SATS equipment for instance. The third is to ask one of the pilots to enter a hold, describing a circle or a racetrack pattern, while the other aircraft proceeds on course (described in the following sections).

For an airport with an ATC operator, a sophisticated display can be designed to assist the operator. This display would show each aircraft location and route, according to data received by the SLF and manual entries from the operator. When a pilot identifies him or herself to the airport, and announces his or her intentions, the controller enters these parameters into the SLF data set and the display provides a route. Moreover, for potential traffic conflicts, it would show the location of the threat, and suggest another route.

For airports without an air traffic controller, we suggest the use of automatic voice communication. This system would work with software to take care of radar response collisions (see Chapter 5 for further details about ATC automation), compute and propose a traffic solution to avoid any traffic conflict.

This SLF device is particularly interesting for GPS equipped aircraft. GPS approach procedures are already more flexible than other approaches. The combination of on-board GPS and SLF will give to the pilots a more precise route, as the location of the aircraft will be more accurate. The air traffic around airports will be less congested and therefore safer.

Now, if the pilots decide by themselves what to do when they detect a traffic conflict (for free flight, for instance), they have to detect the conflict, decide and act. To detect the conflict is easy with TCAS. For non-TCAS equipped aircraft, the only solution available today is visual detection. Under VMC conditions, visual detection is possible since arriving aircraft will fly around the airport approximately at the same altitude. The SATS program will extend free flight to all weather conditions. Therefore, SATS cannot rely upon visual detection alone. Once the conflict is detected by the SLF on the ground, the information will be broadcast to the pilots by automated voice.
The limitation with this system is the following: what is going to happen if the incoming aircraft is not SATS equipped and is listening to Roanoke airport? In that case, the two airports need to coordinate their traffic information.

4.4.2 Two aircraft are flying toward the same point

The air traffic controller can establish priorities in the order North, West, South, and East, for example. For instance, an aircraft flying from West to East wants to reach the IAF, as the same time, another aircraft flying from North to South (Figure 4.5). Since North has priority over West, the first aircraft (West to East) will have to wait two minutes in order to avoid data collision (see previous section), while the other one reaches the IAF and begins its approach.

Figure 4-5  (a) Initial Conditions: aircraft A and aircraft B at the same distance D

(b) Modified Maneuvers: Aircraft B (North) has priority over Aircraft A (West) and proceeds. Aircraft A in 2min. hold

A standard rate turn for a General Aviation (GA) aircraft is a 360 degrees circle in two minutes. A two minutes delay is also the time delay needed to avoid data collision between two transponder responses.
Now if a third aircraft is coming from the West while the first one is completing a hold, assuming that the third aircraft detects the first one and understands the procedure, it can also enter a hold until the other aircraft are clear of the approach path (Figure 4.6). Such a condition can occur since the SATS goal is to reach one movement every 5 minutes.

**Figure 4-6**  A third Aircraft C detects Aircraft A, and begins a holding pattern. Aircraft B proceeds while Aircraft A is holding. When Aircraft A finishes the holding pattern, Aircraft A proceeds. When Aircraft C finishes the holding pattern, Aircraft C proceeds.

Another solution consists in letting the airport ATC controller detect the potential conflict, assign the priorities and communicate the instructions to the pilots.

Instead of holding, the second priority aircraft could modify its approach, in order to pass behind the first aircraft and to give it enough time to the first aircraft to land (Figure 4.7).
The airport of Blacksburg, VA can be used as a model for small airports. Its IAF is about 24 Km away from the airport. The two aircraft need to keep a safety distance of 3.1 Km, and/or to be seen with an angle wider than six degrees (for a six degrees beam-width antenna) from the ground sensor. Figure 4.7 describes the figures to use for this approach. With a small angle (six degrees), and a large distance (24 Km), the arc centered on the airport looks like a rectangle. If the second aircraft turns around this rectangle, no data collision will occur. This rectangle’s dimensions are 3.1 Km and $24 \times \tan(6^\circ) = 2.5$ Km. With a safety margin, the dimensions of the rectangle may be $5 \times 3$ Km ($2.7 \times 1.7$ n. m.). This pattern is an alternative for a hold.

**4.4.3 An aircraft takes off while another one is completing a missed approach procedure**

The missed approach procedure typically consists in turning to a specified heading and climbing. When an aircraft takes off, the pilot may want to fly a
heading opposite to the runway heading used for take-off. What happens if another pilot making an instrument approach is flying the missed approach?

In order to detect the two aircraft using a TCAS unit at the airport, ATC has to give a two minutes delay between the arrival time of the first aircraft in approach, and the time of departure for the second aircraft. If ATC knows when the first aircraft reaches the IAF, it can compute the time when it will know if the landing is successful or if a missed approach procedure is being flown, and therefore it can give an approximate time for departure. For instance, an aircraft wanting to fly a GPS approach into Blacksburg reaches the IAF at time $t$, and a second aircraft receives its heading from the ATC. The first aircraft should land at time $t+\Delta t$, therefore ATC asks the second aircraft to be ready to take off at approximately time $t+\Delta t+2$ minutes. The time needed to complete the landing procedure is nine minutes after the aircraft reached the IAF (at 24Km, for an aircraft flying at 150Km/h is $\Delta t = 9$ minutes). During these nine minutes, the arriving aircraft may announce a missed approach. Consequently, the departing aircraft should be ready to take off at $t+\Delta t+2$ minutes, but should also wait for the landing confirmation of the other aircraft.

4.4.4 Pilots practice in the airport traffic pattern while another pilot wants to land

To train, pilots take off, turn around the airport before landing and eventually repeating the operations several times (Figure 4.8). Before turning from one segment to the following one, the pilot checks visually if any aircraft are on the segment he/she wants to begin. This way, there can be no more than one aircraft at the same time on a single segment. But, when an aircraft is approaching the airport, its pilot has to detect training aircraft. Currently, this detection is visual; but it is not a problem for the arriving aircraft, since the pilot comes from a higher altitude, and has a good vision of the airport airspace. On the other hand, it is not easy for a training pilot to detect incoming aircraft visually. If pilots are SATS
equipped, they will be able to detect all traffic in the pattern, and each incoming aircraft.

Another issue is the time during which the incoming aircraft transponder response will be overlapped by training aircraft transponder response. Since both of these aircraft are to land, and operate in a small portion of the airspace around the airport, their transponder responses are likely to be overlapped. To limit this situation, FAA can make sure the training aircraft altitude is below the IAF altitude, and the IAF is far outside the training pattern: at least 3.1km away from the pattern. Actually, this configuration is currently used in all airports.
4.5 New procedures simulation

The approach to the question of minimal separation for SATS equipped pilots so far has been very analytical. We looked at specific conflicts and proposed solutions. These solutions work according to computations. The experiment designed to investigate the solutions is Eric Shea’s simulation [Shea, 2002]. The simulation’s environment describes an airport with a single runway, and airspace divided into two portions: the first one for arrivals and the other one for departures (Figure 4.9).

![Airspace layout in Shea’s simulation](image)

**Figure 4-9** Airspace layout in Shea’s simulation

In Shea’s simulation [Shea 2002], the arrivals and the departures occur at random times, with random directions. For our problem, we need to modify this simulation in order to use specified times and directions.

For a 490 sectors SLF antenna (i.e. a purely ideal beam width narrower of 1 degree), with random time and random directions of arrival and departure, the
percentage of data collisions for 40 movements per hour is 70%. With only 120 sectors (a three degree beamwidth), but specific time (a three minutes separations between each arrival or departure) and directions (no crossing path), there are no collisions with 40 aircraft per hour. This dramatic drop is due only to traffic organization, using the figures computed in this chapter.

4.6 Simulation results

Shea’s Matlab simulation [Shea, 2002] represents an airport and its airspace. Aircraft may arrive at random time, from random directions. Once they have appeared in the airspace, they fly toward the Initial Approach Fix (IAF), before following a glide slope and landing. Similarly, aircraft may take off at random times and fly toward the Standard Instrument Departure point (equivalent to IAF, for departures) before flying away following a random heading. Shea’s simulation has been used for SSR statistical studies [Shea, 2002]. In this section, we are not really interested in statistics, but more in case studies.

Shea’s simulation has been modified in order for the user to specify the time of arrival/departure and the initial azimuth/heading. This way, the user may create a conflict. Furthermore, when a future conflict is detected, the aircraft involved are to modify their flight path according to the rules we set up. On Figure 4.8 one can see the simulation results for the situation described in section 4.4.2. Two aircraft are heading to the same point (the IAF). The aircraft A proceeds and flies toward the airport while the aircraft B describes the path described in section 4.4.2. With the square dimensions equal to 5 × 3 Km, the simulation shows no data collision. Therefore, the new maneuver described in section 4.4.2 separates the two aircraft sufficiently to avoid any conflict, and to detect both aircraft all the time.
Concerning the flight path described in section 4.4.4, the standard departure point has been set 1Km away from the airport, whereas the IAF for all incoming aircraft remains at 15Km (Figure 4.11). In that case, the maximum outage time (maximum number of consecutive overlapped transponder responses) for a four sector antenna is 94 seconds, for an antenna with 45° resolution the outage time is 69 seconds, and only 37 seconds for a 3° antenna beam width. This outage time corresponds to a training aircraft and an arriving aircraft landing at the same time. This case, obviously, will not be permitted, and a minimal separation will be demanded. If the arriving aircraft is SATS equipped, its pilot will correct his or her flight path. But if this aircraft is not SATS equipped and is listening to ATC at another airport, Roanoke for instance, there is no way for the SATS system to alert the incoming aircraft. Moreover, for cloudy conditions, the incoming aircraft will come out of the clouds and the training aircraft will not be able to detect it visually. If the training aircraft is SATS equipped, it will detect the arriving aircraft and follow an emergency procedure.
This particular problem may be solved by traffic information coordination between ATC systems at the two airports: Blacksburg and Roanoke. If the ATC systems at the two airports exchange information, they will be able to alert the two aircraft.

### 4.7 Conclusion

“TCAS-on-a-stick” as part of the SLF reduces the delays between movements around the airport (arrivals and departures) and avoids traffic conflicts. Nevertheless, its use demands an adaptation of current procedures. TCAS-on-a-stick requires new traffic organization, holds, and priorities. The
presented simulation has shown the new traffic organization can handle shorter delays in small airports, and continuously detect aircraft around the airport for twenty movements (arrivals and departures) per hour. Nevertheless, a SATS limitation has been identified: when an aircraft is not SATS equipped and listens to ATC at another airport while beginning its approach to an airport, this pilot cannot be alerted in case of traffic in the pattern. Now, with reduced delays and increased number of aircraft, the risk is to overload human controller capacity, especially in a small airport. Is a fully automated ATC possible presently?
Chapter 5 - Smart Landing Facilities Automation

Impact on current ATC, issues, and potential implementation with expert systems

As the demand for air transportation increases, efficient Air Traffic Control (ATC) is required. We have seen in the previous chapters how to upgrade the equipment at small airports, and to decrease the delays between movements on the runway. Nevertheless, human stress and limited workload make a boundary to more productive ATC. The ATC controller is responsible for separating aircraft safely in a volume of airspace. Instructions for heading, altitude and speed are issued by voice radio, which all pilots in that airspace are to listen to, and to which all pilots in that airspace are to respond by voice. The controller then watches his/her radar screen to make sure the pilots follow the correct instructions. With the current system, each one of these steps requires the full controller attention. For small airports, as well as for congested airspaces, ATC automation can lighten the controller workload. The FAA claims ATC is 99.999% error free. This is probably close to the truth. That is a far better system than almost any other, so any replacement must be as good, or better, and must take into account future US air traffic evolutions. The ATC decision-making is today hardly automated, and the “controlled subjects” - the pilots - are humans. Therefore, instead of looking directly at the needs and trying to implement a solution, it seems quite natural to have a human-centered approach for this problem. We will see the impact of automation on ATC human operators, and the potential issues, coming from human-machine interaction. Since automation is now demanded to maintain the same quality of service, and since the slightest step toward automation will deeply involve Human-Machine Interaction (HMI),
our discussion will start with HMI in ATC. We will see first how the automatic system can work, then if new equipment will be needed, and finally how close to the current system the new automatic ATC can be designed. Aviation is a field very reluctant to change because safety of life is involved. We will see that an unreasonable and sudden change to full ATC automation would avoid human limitations, but would also take human controllers “out of the decision loop”. FAA will not accept that. If a change is necessary, it needs to be implemented slowly, nearly without any noticeable change. Finally, we will see the future requirements systems for ATC, and we will see that Artificial Intelligence matches these requirements. Among the abilities demanded in ATC are judgment, reactivity and adaptability. Recent improvements in Artificial Intelligence and expert systems offer these abilities.

5.1 – Impact of Automation on ATC controllers and issues

5.1.1 – Arguments for partial or full automation

Human stress and limited workload demand automation’s help for ATC decision making

As the US air traffic increases, the need for more efficient ATC is obvious. In VMC (Visual Meteorological Conditions), the localization of one aircraft by another is visual, and identification is nearly impossible. According to a study done by the National Transportation Safety Board (NTSB), the result of the current system was 540 midair collisions involving general-aviation aircraft in the 20-year period from 1960 to 1979 — an average of 27 per year or, about one
every two weeks. That rate had dropped to 15 per year in the 1990s (or one every four weeks). There is agreement that increase in demand will eventually exceed the current systems’ capacity. As a consequence, either air traffic will become overflowed and will reject any increase, or ATC will soon not be able to satisfy the safety demanded by the Federal Aviation Administration (FAA).

Moreover, the human stress and workload will continue to go up until reaching the end of human capabilities. Finally, automation has made great progress, especially in the past ten years, with artificial intelligence.

An expert system will avoid most of human errors

The most effective way to design an automatic system for ATC begins with understanding human’s most common errors. 90% of these errors can be divided into three categories (Stager, 1991): attention, communication and judgment. Among the errors related to these categories, the most common are non-recognition of conflict, inattention, deviation from required operational procedures, failure to communicate and poor judgment. An automatic system has many advantages over human beings, and does not have these sources of errors.

Communication: Since automatic systems will work with pre-recorded sentences, and non-ambiguous instructions, communication cannot be a problem. On the other hand, transmission may be a problem if the message is not correctly delivered.

Attention: Human can forget an instruction, an aircraft or a conflict; this situation cannot occur with an automatic system, if enough work memory is provided.

Judgment: To compare judgment skills of an expert system to those of a human being, experimentation and further investigation are necessary. Nevertheless, let us note that in many applications, expert system cognitive
abilities went beyond human skills (e.g. Big Blue beating chess master Kasparov).

5.1.2 – Issues related to human-machine interaction in ATC

Nowadays, automation is partially used for ATC, especially to obtain, store, compile and condense the traffic information. It is possible to provide a system that analyses the airspace situation and gives a solution to the human controller. Then, if the controller approves the solution, the system automatically takes care of the transmission of instructions to the pilots. Such a system is presented in the following paragraph.

Therefore, the immediate question is: can we replace the human controller in the process? A human controller, with such a system, must get used to adopting the system’s solution without understanding it and without systematically checking it (Hopkin, 1991), and this is unlikely to be accepted by the FAA. In this process, the human action will be restrained to pressing a button. Hence, the system will compute a solution; this solution will be nearly systematically adopted by a human controller, and the system will implement it. If the only task of a human controller is to push a button, can’t we keep only the automatic system?

The only reason why we would keep a human being would be for a manual reversion. Now, when all the controllers will have lost all experience of computing air traffic solutions, no controller will take a chance to do a manual reversion. Otherwise, the controller would have to be correctly trained to do a reversion (that he/she would do rarely), and will follow all previous maneuvers to understand the problem. This is simply not feasible for a human being. No one can ask a controller to adopt 99.999% of the system’s solutions - safety figure necessary for any system to be adopted by the FAA – and, at the same time, be ready to take back control. Obviously, if a system does not allow human controllers to take over in case of failure, the FAA will not accept this system.
Therefore, the human-machine interaction cannot go beyond a certain point. If we go too far in automation, no human controller can be accepted in the system. Nevertheless, we will see later that there is a role, which has to be done by a human operator.

Therefore, automation has to be adopted fully or has to stay within a certain band, to be determined. And if it is fully adopted, the immediate issues raised are the questions of trust from the pilots in the automatic system, and the responsibility in case of failure. Nevertheless, let us notice that small aircraft collide flying under VFR rules at a rate of about 15 per year, and ATC is rarely controlling these planes. On the other hand, there has been no collision between passenger aircraft controlled by ATC in the US in the last 25 years. Therefore to improve the current system, FAA needs to reform VFR rules and develop ATC for small airports. This new ATC for small airports needs to be as good as, or better than, the current ATC system. We will see later under what conditions automation can be fully adopted, and when an operator has to be present.

5.2 – Impact of Automation on current ATC systems

We have seen how automation would help, in terms of workload and stress – or productivity and safety. We have seen how the airspace traffic should be organized in order to achieve the best performance, in both safety and number of movements (departures and arrivals). The next question is: can this system be automated without changing the current procedures and equipment? And how?

To answer these questions, we need to analyze the ATC decision process and to decide what can be automated. There are different levels of automation - acquisition, data presentation, decision-making and solution implementation – and each one of these can be separately automated. How will that affect human
controllers? Finally, automation needs to stay very close to current systems in order to be accepted by the FAA, the ATC controllers and the pilots. Therefore the future system needs to be built over the current system, before being eventually modified forward a fully automated system.

5.2.1 – Description of the decision process

We need to take into account several agents of decision: the pilot, the air traffic controller(s), the instruments on-board, and the instruments on the ground. Primarily the pilot can make the decisions - in a pilot-centered system like uncontrolled VFR approaches - or an automatic ATC system can make decisions. In a pilot-centered system, all the available information has to be brought to the pilot, according to his/her own instruments (Mode C transponders, TCAS, ADS-B, etc). If we decide to keep a system where air traffic is globally arranged by ATC, for each aircraft, we need to automate the decisions made on the ground, and to provide a data link between ATC and the aircraft. The data link will have to be designed for the minimum required equipment. Then, another data link for optional equipment can eventually double upon this one: TCAS and ADS-B for instance. This solution has the great advantage to provide coherent standard instructions to each aircraft. Many accidents occur because of incoherent instructions between the on-board instruments and ATC on the ground. This solution allows also further improvements, depending on each aircraft’s equipment.

Therefore, one description of the decision process consists in several steps shown on Figure 5.1.

A “TCAS on the stick” system (Shea, 2001) is a Smart Landing Facility (SLF) device, working like a Traffic Collision Avoidance System (TCAS) interrogator/transmitter. The TCAS is optional for small aircraft, but is required by the FAA for commercial aircraft. We can distinguish the aircraft present in the airspace into two main categories: TCAS equipped aircraft and the non-TCAS
equipped aircraft. This last kind of aircraft still has a transponder—required by the FAA. During the introductory period of our system, all aircraft will not be fully equipped. Therefore, we need to provide a secondary system, based on Secondary Surveillance Radar (SSR) technology.

![Diagram](image)

**Figure 5-1** Decision process for traffic instructions

With these two systems, we are able to detect and identify each aircraft. Analyzing these two systems’ information, a third device computes a solution for each aircraft and broadcasts it. The TCAS-equipped aircraft can receive this information digitally and send it to the pilot throughout a display. The non-TCAS aircraft’s pilot can receive the information through voice communication.
5.2.2 – Different levels of automation: data acquisition, data presentation, decision-making and solution implementation

The basic forms of automation are more quantitative than qualitative: gathering, storage of data, compilation, presentation of data. Therefore, the easiest way to use automation in ATC, without any big structural change, is in acquisition and data, while the controller does the analysis and solution implementation. Concerning the acquisition of information, automation can help by using better sensors, such as “TCAS-on-a-stick” or monopulse SSR (Stevens). Concerning data presentation, the current system has taken advantage of recent improvements in the Human-Machine Interface. It is very important to continue forwards a more efficient data presentation. For instance, it would be useful to label each radar “blip” with target’s speed, heading, altitude, and ID as the current system already does, but also with clearances and instructions. But if the automation does not go further, the aircraft separation would still be the controller’s task. The automation would only help the controller with the acquisition and data presentation, and not by making any decision. The ATC operator’s workload would be lighter, but not optimal.

Because of the advances in automation, we can now develop a system to compile a solution, to implement it, and also to give probabilistic information about the future aircraft’s location. Therefore, not only the reception of traffic information on the ground would be automatic, but also, the analysis and the implementation.

For an automatic reception on the ground, we need sensors. As shown above, the system is required to have at least two sensors to work with both “TCAS-on-a-stick” and regular transponders. Moreover, we need to combine these two systems to come up with a solution for the entire airspace. In the future, if we want to add another optional equipment – ADS-B for instance – we will just have to add another sensor.
Then comes the solution’s implementation. In a fully automated system, this implementation will also be automatic. As SATS-equipped aircraft may eventually be present in the airspace, the system can use a TCAS data link to provide SATS-equipped aircraft with instructions. These instructions will then be transmitted to the pilot through a display in the cockpit, with appropriate symbols and color codes. For non SATS-equipped aircraft, the most convenient for pilots, and the easiest to implement without new on-board equipment, is synthesized voice communication. This voice communication will simply work like current human voice communication between ATC controllers and pilots. Pilots listen to this voice radio for their own instructions, and also for the other pilots’ directions. Let us note that this voice communication will just repeat what the SATS-equipped aircraft will receive digitally. Nevertheless, SATS-equipped pilots will listen to the voice communication. Radio oral instructions will not keep them as busy as the display visual information. The display will be an information redundancy. Coordination between voice communication and display is easy to implement. The aircraft following voice instructions will be blinking on all SATS displays in all SATS-equipped aircraft cockpits, and its trajectory will be temporary shown with a dotted line.

5.2.3 – Automation for controlled and uncontrolled approach

The present traffic organization varies, for uncontrolled or controlled approaches, with VMC conditions or IMC conditions. Let us see for each one what is the current system, how an automatic ATC may be implemented, and what would be its impact for pilots and ATC controllers. Pilots are reluctant to change. Therefore, even if automation needs to be implemented, the overall ATC system cannot change dramatically, and pilots should not see the change.
Uncontrolled VFR conditions

For uncontrolled VFR conditions, the pilot is required to make a set of radio announcements depending on where he/she is. The approach path to an airport requires a 45° entry into the airport traffic pattern, and announcements are required at each turn. As described on Figure 5.2, for uncontrolled VMC conditions, the aircraft is required to fly a certain path. At each turn, the pilot has to check that no other aircraft is present in or approaching the next segment, and to announce his or her intentions over the airport’s CTAF (Common Tower Advisory Frequency). Incoming aircraft usually fly a 45 degrees approach to the airport traffic pattern. With this procedure, the pilot has to make a couple of checks before any maneuver, and the possible locations for an aircraft are limited.

The main problem with this system is the detection of other aircraft by each pilot, since VMC conditions allow a visual detection. First, an aircraft is rarely where it is supposed to be, or more precisely, where the other pilots think it is. Second, even if an aircraft is detected, the pilot can never identify the aircraft with absolute certainty. All pilots listen to the announcements on the radio to determine approximately the location and heading of the other aircraft. For safety, an automatic ATC must provide the same, or better, capability. With the current system, pilots keep a mental picture of the traffic. This keeps them busy, and they can miss a radio call, or form a wrong mental picture. To assist the pilot, SATS aircraft have a visual display that shows the situation. This display keeps a basic code to show other aircraft headings, altitude and ID and is quickly readable by the pilot.
Figure 5-2  Pilot radio announcements for approach in uncontrolled VFR conditions at an un-towered airport. Rectangle shows standard left hand traffic pattern for VFR arrivals and practice.

For VFR uncontrolled conditions, the automatic system will guide each aircraft with instructions, make the public announcements, and foresee potential conflicts – eventually asking for holds and assigning priorities. Incoming aircraft will be detected as soon as they penetrate a designated distance from the airport. Therefore, the two problems – detection and identification – are solved, while the public announcements remain in place. Moreover, the backup system for this automatic ATC can simply be the former manual system. The conversion from automatic to manual system would be very easy.
IMC conditions

For IMC conditions, pilots are required to fly the approach given in the approach plates. This approach may vary from one airport to another depending on local conditions. Nevertheless, the procedures for IMC approaches are not very different from those for VMC conditions, in terms of detection and instructions. Each pilot is required to identify him or herself and to announce his or her intentions. But the issues of this manual system (localization and identification), and the benefits brought by an automated system remain the same. Actually, an automatic system would take care of the different approach types (IMC and VMC).

For both uncontrolled conditions, the distance required between two aircraft, landing or taking off, determines the number of movements per hour. This distance has been computed to minimize the risk of collision, taking into account a large margin for human error and detection delays. The use of an automatic system would greatly reduce this margin. Ideally, the minimum safe distance would be entirely determined by the sensors’ capacity on the ground: three minutes between two aircraft. (This figure is only valid for small aircraft, since other parameters may appear for larger aircraft, like wave turbulences created by the leading aircraft).

Controlled movements

For controlled movements, each pilot receives his or her own instructions through airport voice communication. The pilot also receives the other pilot’s instructions in the same way. These instructions are (non exhaustive list):

- Cleared for landing / take off …
- Fly heading … degrees
- Climb / descend to … ft
- Caution traffic … o’clock, altitude … ft
Expect … climb/descend to … ft, in … minutes

Vectors to IAF

If an instruction is not clear or doesn’t correspond to the pilot’s intentions, he/she can ask the controller to repeat or to correct the instruction. Most of the ATC detection is automatic. The controller has to analyze the information presented by the system, compute a solution and implement it.

The issues with this system are related to human performance. Human workload and tolerable stress are limited, and determine the number of movements allowed as well as airspace safety. Despite the increase in demand, the human workload capability remains the same. Therefore, either the number of movements must the same, or safety will decrease.

To increase safety and the number of movements allowed, the SLF has to provide a better sensor, better data analysis, and a more efficient way to implement the solution. To go beyond human workload, it is necessary to use automation, at least for the sensor, data analysis and the information presentation. The question remaining is: will we keep a human controller for implementation? The factors to take into account are presented above, in the first paragraph. We can use an automatic system to present the potential solution, but we need to explain this solution to the controller, and not only to present the result. Otherwise, the task of the controller will be reduced to pressing a button, and he/she will never be able to make any manual reversion. Therefore, if we go beyond that point, if we allow the system to design solutions and to implement one of them, full automation is required. The human controller’s role will be to monitor the system, to help going from a first step of automation, to gain pilots’ trust, for the second step: full ATC automation.

For each of these conditions, the automatic system will remain very close to the current system, helping pilots to accept it. The only new on-board equipment required is an eventual display for SATS equipped pilots, but this equipment is not necessary initially. On the ground, new sensors are required for better localization of aircraft. The sensor system proposed in this discussion is based on the “TCAS on a stick” system [Shea, 2002]. The ATC software will be
updated by the automatic system computing solutions and implementing them. This new machine can be easily updated for future improvements, since only new software has to be downloaded.

5.2.4 – Double use of TCAS plus ADS-B

ADS-B system is particularly useful in congested airspaces. TCAS systems use an interrogation-response protocol. Therefore, electromagnetic pollution increases exponentially as the number of aircraft increases. On the contrary, ADS-B aircraft simply broadcast their position and identity. Therefore, electromagnetic pollution simply increases linearly with respect to the number of aircraft.

In a congested airspace, instead of relying entirely on the TCAS system, and broadcasting the same information to each aircraft separately, we can broadcast it to an ADS-B equipped aircraft. This aircraft will then broadcast it to all other ADS-B equipped aircraft.

5.3 – Artificial Intelligence for ATC automation, a potential implementation

We have seen how an automatic system could lighten the controller’s workload, and how it would be implemented. We have seen how close to the current system full automation would be. We now have to look at the required functionalities of a future automatic ATC system. Then, we will see that Artificial Intelligence matches these requirements pretty well, and we will propose an implementation.
5.3.1 – Requirements and functionalities of an automatic ATC system

The decision process will remain the same: data acquisition, information analysis, decision-making, implementation and error-control. Therefore, we can propose a list of functionalities for the future system:

- Regular aircraft interrogation (localization, identity, intentions)
- Information Reception
- Analysis of several data formats
- For a new aircraft in the airspace: computation for a flight plan based upon heading and distance.
  One way to implement it, would be to ask the pilot: “if you do not plan to land, make a xxx degrees turn and fly toward heading yyy degrees for zzz minutes”
- For an aircraft already known: comparison of location with flight plan, error-control (generation of alerts) or new instructions
  Future airspace state prediction
  Error control

To these functionalities, we need to add a way to evaluate and grade the performance of the automatic system. One solution consists in keeping the same evaluation for human and automatic systems. We also need to provide an auditing system, and failure detection. This auditing trail cannot be provided only by the audited system, and the auditing trail needs to be protected against the audited system’s failures.

Another functionality may be added: self-improvement. Optimum performance of an automatic ATC system demands that this system learn and modify itself in order to prepare new concepts, new operations, and solve new problems. This self-improvement does not have to be understood as anticipation, but adaptation. Human controllers have the skill to adapt themselves to new
conditions (or at least partially, and under certain circumstances and constraints). If we want full automation to replace current human operations, we need to keep this skill. Recent progresses in Artificial Intelligence have demonstrated this ability for certain types of systems.

5.3.2 – Object-oriented languages and programmed computing systems for ATC. The need for cognitive engineering

Before studying a potential application of neural networks (see next section for definition) to ATC, we have to underline the fact that other solutions have already been tested:

Hojong Baik has been studying the use of object-oriented languages for ATC, and especially the Aircraft Separation Problem [Baik, 2000].

The Dutch National Aerospace Laboratory has been working with the Center Terminal Radar Control and the NASA Ames Research Center on a simulation of fully automated ATC [AGARD, 1993 chap. 12-1]. They have used a programmed computing system.

The project AERA 2 leads to a system based upon conflict detection, 20 minutes in advance, an automatic solution proposed to the controller, simulation possibilities [J.C. Celio, 1990].

But all these projects have provided systems where human beings are absent in the decision-making. The only task left to the human controller is the validation of the system’s solution. Therefore, as we saw in the first paragraph, the controller will lose all experience in this scenario. The Human-Machine Interaction will be cut; Machine will have a prime role over Human [Leroux, 1991 & AGARD, 1993 chap. 15].

For a human controller to stay effective, the decision-making cannot be left fully to the machine. On the contrary, the automatic system and the human controller have to work complementarily. Therefore, the system has to adopt the same cognitive model than humans, plus error detection.
5.3.3 – Use of a neural network for automatic ATC systems

What is Neurocomputing?

Neurocomputation is a data – or information – process (Ham, Kostanic, 2001). Its main applications are pattern recognition and optimization. Unlike programmed computing, Neurocomputing implies a learning process. The neurocomputed system has first to learn how to solve a certain problem before generalizing not the solution, but the solving process itself. During the learning step, the system learns how to recognize a problem, before learning how to solve it or to optimize the solution. Because of this learning process, neural network systems are used where adaptability is required. Another feature of Neurocomputing is its robustness. Because of its very nature, a neural network is a distributed problem-solving system. Therefore multiple failing points are required for the overall system to fail.

How Neurocomputing can help in ATC automation?

Let us review the decision process: data acquisition, information analysis, decision-making, implementation and error-control. Regular programmed computing can easily be used for data acquisition and implementation, since data formats and equipment barely change. If a change is needed in the future, a software update will be performed, but the low frequency of change in ATC does not justify the use of Neurocomputing for data acquisition and implementation.

On the other hand, decision-making can fully exploit Neurocomputing advantages. The adaptability feature and the use for solution optimization have been described above, but there are many others. Among them, statistic estimation can be a great use of Neurocomputing for ATC. Since aircraft
arrivals/departures are not 100% random, statistic estimation can be used for prediction, anticipation, and adaptability.

The neural network approach for ATC automation is far from being science fiction. The ERATO project (Leroux, 1991) has successfully demonstrated neural network can be used in an ATC system, where human controllers have a lighter task, but are still needed. In this system, they focus all their attention on deviant aircraft and potential conflicts, while the system takes care of all other aircraft, presents to the controllers the risks, and simulates the solution proposed by the human being.

5.4 The role of human in automation: system manager, teacher. Future improvements

We have seen that automation is becoming more and more necessary, as the demand for more efficient and secure ATC increases. We have seen that automation cannot go beyond a certain limit, after which human controllers’ tasks would be so reduced that they would be put “out of the loop”, and become unable to make any manual reversion.

Then, automation can be used at each level of the decision process. Automation is also highly compatible with the “TCAS on a stick” SLF sensor. Moreover, automatic systems can remained very closed to current procedures, for both controlled and uncontrolled current approaches.

Finally, the functionalities that would be asked to an automatic ATC system are very closed to what neural network expert systems can offer. A possible implementation can therefore use a neural network.

What is the future task of human in ATC? All expert systems need supervisors. The human controller may focus his attention on system manager’s tasks: to check, ask for other options, and eventually valid the automatic
system’s solutions. Human has the great ability to talk and understand human. Therefore, it will be necessary to keep human controllers to look for optional solutions, and to provide assistance to pilots. So in this scenario, the decision to follow such or such option will be taken by the human controller, but computed and implemented by the automatic system.

The next step of automation will be an automatic system collecting data more efficiently, computing solutions, implementing Human’s choices, and automatically adapting itself to new environments. The step beyond will see the human controller as a system manager: the controller will not have a role in the decision itself. Improvements in automation will then allow automatic pilot. At that point, the distributed problem-solving subject will have a dramatic importance (Cammarata, McArthur, Steeb 1983). Nowadays, no totally satisfying solution for distributed problem-solving in ATC has been found. When distributed problem-solving and expert systems cooperation will reach sufficient levels, larger scale – eventually nation-wide – ATC automation will then be allowed.

In terms of safety and efficiency (number of movements per hour), the upper limit for automation will not be the current stress/workload limit, but the equipment limit. This equipment limit is defined by the SSR and security limit: a 2 minutes separation between aircraft is required. With automatic pilot, and eventually data link improvements, these limits will no longer exist.
Chapter 6 – Conclusion

A more efficient ATC for small airports is possible with current technology. A better ground sensor can track aircraft in the airport airspace with accuracy. This information needs to be available to all pilots in order to allow free flight. The SATS program proposes a Smart Landing Facility for better aircraft tracking and traffic information broadcasting to pilots. Once traffic information is available to all pilots, a set of strict procedures has to be designed to coordinate the traffic and get the maximum efficiency from ATC. Overlapped received signals can be separated, as described in section 6.1. The use of TCAS as a ground sensor, and a potential application are presented in section 6.2. The issues raised by a fully automated ATC are summarized in section 6.3. And finally, section 6.4 proposes future investigations.

6.1 Overlapped Radar responses can be separated

The main limitation of secondary surveillance radar systems is the risk of seeing overlapped radar responses if two aircraft are within a 3.1 km range band from the ground sensor. A solution consists in separating the two responses. Chapter 3 presents a method to achieve that.

Each radar response can be characterized by a set of parameters (pulse width, frequency, and initial phase). With accurate instruments, it is theoretically possible to collect these parameters, and therefore, to separate each bit. When a potential solution is found for the two responses, the system rebuilds the original signals with the set of parameters, and the sum of these rebuilt solutions is compared to the original received signal. If the sum corresponds, the exact solution has been computed; if it does not match, the system changes the bits.
that create a problem and compares the sum again. Such a loop can solve most of the overlapping problem. But it is not necessary to recover all the overlapped signals, as long as the system recovers most of them, and does not make a large number of consecutive errors. If after a certain time to be determined, the loop still does not give a good result, the received signal is thrown away and the aircraft may disappear off the radar display. For a 2Hz Radar (2 interrogations per second), one failure to separate the radar responses means the radar system waits $\frac{1}{4}$ second until a new response comes. The present system simply consists in throwing the packets away when responses are overlapped. What is important is to avoid throwing away a large consecutive number of packets, because the aircraft implicated in the data collision could disappear off the screen. The new separation technique recovers most of the collided packets, and can be used as a backup system for current ATC radar systems.

Nevertheless, with amplitude differences in the received signal from different transponders (chapter 3 showed that case for a 10dB difference), some errors may not be solved. Further investigations are required to determine what is the critical amplitude difference.

### 6.2 The use of TCAS as a ground sensor

TCAS used as a ground sensor, without any attempt to separate overlapped signals’ is not really useful. In a Matlab simulation describing an airport airspace divided into two zones (one for arriving aircraft, and the other one for departing aircraft), 20 aircraft landings or take offs lead to more than 30% of data collisions, and a maximum outage time of 450 seconds’ according to Eric Shea’s simulation [Shea, 2002]. But if one assumes that two overlapped responses can be recovered, the probability of a triple collision is insignificant, even for 60 movements per hour (1 movement every minute).

Moreover, “TCAS on a stick” uses existing technologies. TCAS is already widely commercialized, and can work with non-TCAS equipped aircraft (with
Mode C All-calls technique [Stevens, 1988]). Nevertheless, the use of this system demands an adaptation of current procedures. TCAS-on-a-stick allows new traffic organization, holds, and priorities. The simulation presented in chapter 4, and modified to take into account different sets of procedures, has shown that the new traffic organization results in shorter delays for small airports.

Nevertheless, some problems may occur if a pilot does not follow the correct flight procedures, and is not equipped with SATS. This limitation demand current procedures be maintained until all GA aircraft are equipped with SATS.

6.3 A fully automated ATC and Human-Machine Interaction

As the demand for more effective ATC increases, ATC automation becomes necessary. But going too far in automation risks putting the human operator out of the loop. Nevertheless, automation can be performed as long as the human operator is ultimately the one responsible for taking the decisions. Moreover TCAS-on-a-stick, and the set of procedures proposed in Chapter 4 are well suited for automation.

The next step of automation will be an automatic system collecting data more efficiently, computing solutions, implementing the human operator choices, and automatically adapting itself to new environments. The step beyond that will see the human controller as a system manager: the controller will not have a role in the decision itself. Improvements in automation will then allow an automatic system to control the aircraft for approaches to all small airports, and take offs. At that point, the distributed problem-solving subject will have a dramatic importance. Nowadays, no totally satisfactory solution for distributed problem solving in ATC has been found. When distributed problem-solving and expert
systems cooperation has reached sufficient levels, then larger scale ATC automation will be allowed.

In terms of safety and efficiency (number of movements per hour), the upper limit for automation will not be the current stress/workload limit, but the equipment limit. This equipment limit is defined by the SSR and safety limit: a two minutes separation between aircraft is required. With automatic pilots, and eventually data link improvements, these limits will no longer exist.

6.4 A proposal for future research: a better data link from ground to aircraft

The SATS program is still at its beginning, and needs further investigations:

- The system proposed in this thesis is pilot-centered. Since free flight in all weather conditions is the goal to be achieved, the information provided to the pilot is critical. Therefore, an efficient data link has to be designed by electrical engineers, and the extra information has to be tested by human factors engineers to see if pilots can use it efficiently.

- A prototype must be designed and tested. This prototype may require changes in the theory. When a prototype is developed, a set of measurements will be done to confirm the theory.

- Other systems using cellular technologies can be studied for better identification, and to avoid data collision inherent to radar systems. If compatible with TCAS-on-a-stick, they may be of a great use for congested airspaces.

- The FAA may not accept the procedures proposed in this paper, which are an attempt to use a ground sensor to allow free flight. Traffic engineers must test them and design new procedures.

- The cost and feasibility of the system has yet to be studied.
- If the system is used nationwide, it will have a great impact on GA flights, on small airports, as well as on hubs, on aircraft manufacturers, on pilots formation and on air travel companies. A statistical analysis by air traffic engineers is required.
Appendix A: Matlab codes

Here is the Matlab code used for the PLL

% PLL

clear all

phase_signal = input('signal phase? \n');
freq = 30e6;
F_sampling = 300e6;
T_sampling = 1/F_sampling;
t_bit = 0.5e-6;

begin=1;

for t=1:t_bit*F_sampling
    input(t) = sin( 2*pi*freq*(t/T_sampling) + phase_signal);
end

load lpfPll_test.mat;
LPF=Num./Den;

Kd = 5;
Kv = 1/2;
% phase accuracy
phase_accuracy = 0.2;
% integration time in the computation of the mean phase
int_time = 0.2*t_bit;

error = 1; % error = 1 if impossible to lock on
VCO(1)=0;
product(1)=0; product(2)=0;
phase_detected=0;
Ve(1)=1;

t=2;
while( t<=0.5*t_bit*F_sampling )
    Ve=filter(Num,Den,product);
    teta(t)=Ve(t);
    VCO(t)= Kv*teta(t);
    product(length(product)+1) = Kd*(phase_signal - VCO(t));
    t=t+1;
end

while( (t+begin-1 < t_bit*F_sampling) & error )
    mean_phase=0; % mean value of Ve over 10% of length(time)
    Ve=filter(Num,Den,product);
    teta(t)=Ve(t);
    VCO(t)= cos(2*pi*(freq+Kv*teta(t))*(t/T_sampling) + Kv*teta(t));
    product(length(product)+1) = Kd*(phase_signal - VCO(t));
    for k=1:int_time*F_sampling
        mean_phase = mean_phase + teta(t-k);
    end
    mean_phase = mean_phase / (int_time*F_sampling - 1);
    if abs( mean_phase - teta(t) ) < phase_accuracy * teta(t)
        phase_detected = 2 * teta(length(VCO)) / Kd;
        error=0;
    end

end
    t=t+1;
end

output = [product; phase_detected zeros(1,length(Ve));error zeros(1,length(Ve)); VCO VCO(length(VCO)) ];

for t=1:min(length(input),length(VCO))
    difference(t)=input(t)-VCO(t);
end

figure(1); plot([T_sampling:T_sampling: T_sampling * length(difference)], difference); title('erreur');
figure(2); plot([T_sampling:T_sampling: T_sampling * length(output (4,:))], output(4,:)); title('VCO');
figure(3); plot([T_sampling:T_sampling:  T_sampling * length(teta)], teta); title('teta');

phase_detected
disp('time needed to lock orkn'); length(teta)*T_sampling
Now, here is the Matlab code for the demodulator:

```matlab
% function demodulation
% input :    a signal, sampled at 150 MHz
%            the main frequency of the signal
%            the sampling frequency
%
% output :   the signal demodulated
%
% Charles Florin 02/30/02 for the CWT
% last updates 05/14/02

function [output] = demodulation(signal, F_sampling, t_bit, length_info, offset)

% do we have to take the signal or its inverse
signe=sum(signal)/abs(sum(signal));

% Calibrate the threshold for the first sequence
max=0;
for t=offset+4*t_bit*F_sampling:offset+30*t_bit*F_sampling
    if t<length(signal) & signe*signal(t) > max
        max=signe*signal(t);
    end
end
threshold_one = 0.6 * max;

% Calibrate the eventual delay
delay=offset;
```
while(signe*signal(delay)<threshold_one)  
    delay=delay+1;  
end

% integration over t_inter  
t_inter = 1e-7;

for n=1:length_info  
    temp=0;  
    demod(n)=0;  
    for t=floor( (3.5*t_bit-t_inter/2+2*(n-1)*t_bit)*F_sampling ):floor(  
        (3.5*t_bit+t_inter/2+2*(n-1)*t_bit)*F_sampling )  
        if t+delay<length(signal)  
            temp=temp+signe*signal(t+delay)/(t_inter*F_sampling);  
        end  
    end  
    if temp>threshold_one  
        demod(n)=1;  
    end  
end

output=demod;
Here is the code for the second kind of demodulator:

% function comparator
%
% input: the received signal
% the number of points between the two signals
% the pulse width
% the sampling frequency
% the two frequencies
%
% output: the signal demodulated
%
% Charles Florin 03/30/02 for the CWT
% last updates 05/14/02

function [demod] = comparator(signal, deltaPts, t_bit, F_sampling, freq_1, freq_2, phase1, phase2, shape_1, shape_2);

% compare the shape of the first bit with the followings, and demodulate the signal
T_sampling=1/F_sampling;
threshold=0.4;
tau=0.05e-6;       % transition time

amplitude_1 = max(shape_1);
amplitude_2 = max(shape_2);

% demodulation of the first signal
for n=1:13
    signal_to_test=zeros(1,length(shape_1));
    % test for first signal's bit = 1 and second signal's bit = 0
for t=1:t_bit*F_sampling
    time=3*t_bit*F_sampling+(n-1)*(2*t_bit*F_sampling)+t;
    signal_to_test(t)=signal(time);
    theory(t)=amplitude_1*(1-exp(-
t*T_sampling/tau))*sin(2*pi*freq_1*(time)*T_sampling+phase1);
end
for t=t_bit*F_sampling+1:2*t_bit*F_sampling
    time=3*t_bit*F_sampling+(n-1)*(2*t_bit*F_sampling)+t;
    signal_to_test(t)=signal(time);
    theory(t)=0;
end
if sum(abs(signal_to_test-theory)) < threshold*sum(abs(theory))
    % test for first signal's bit = 1 and second signal's bit = 0
    demod(1,n)=1;
elseif max(signal_to_test)<0.2
    % test for first signal's bit = 0 and second signal's bit = 0
    demod(1,n)=0;
else
    test=1;         % test hasn't succeeded yet
    for delta=1:2*t_bit*F_sampling
        for t=1:delta
            test_shape_added(t)=0;
        end
        for t=delta+1:min(t_bit*F_sampling+delta,2*t_bit*F_sampling)
            time=3*t_bit*F_sampling+(n-1)*(2*t_bit*F_sampling)+t;
            test_shape_added(t)=amplitude_2*(1-exp(-(t-
delta)*T_sampling/tau))*sin(2*pi*freq_2*(time)*T_sampling+phase2);
        end
    end
    for t=min(t_bit*F_sampling+delta,2*t_bit*F_sampling)+1:2*t_bit*F_sampling
        test_shape_added(t)=0;
    end
end
test_shape_added(t)=amplitude_2*exp(-(t-t_bit*F_sampling+delta)*T_sampling/tau)*sin(2*pi*freq_2*(time)*T_sampling+phase2);

end
if sum(abs(signal_to_test - test_shape_added - theory)) < threshold * sum(abs(signal_to_test))
% test for first signal's bit = 1 and second signal's bit = 1 shifted
demod(1,n)=1;
test=0;
break;
end
if sum(abs(signal_to_test - test_shape_added)) < threshold * sum(abs(signal_to_test))
% test for first signal's bit = 0 and second signal's bit = 1 shifted
demod(1,n)=0;
test=0;
break;
end
end
if test
for delta=1:t_bit*F_sampling
for t=1:delta
  time=3*t_bit*F_sampling+(n-1)*(2*t_bit*F_sampling)+t;
  test_shape_added(t)=amplitude_2*(1-exp(-(t-delta+t_bit*F_sampling)*T_sampling/tau))*sin(2*pi*freq_2*(time)*T_sampling+phase2);
end
for t=delta+1:2*t_bit*F_sampling
  time=3*t_bit*F_sampling+(n-1)*(2*t_bit*F_sampling)+t;
  test_shape_added(t)=amplitude_2*exp(-(t-delta)*T_sampling/tau)*sin(2*pi*freq_2*(time)*T_sampling+phase2);
end
end
end
if sum(abs(signal_to_test - test_shape_added - theory)) < threshold * sum(abs(signal_to_test))
    % test for first signal's bit = 1 and second signal's bit = 1
    demod(1,n)=1;
    test=0;
    break;
end
if sum(abs(signal_to_test - test_shape_added)) < threshold * sum(abs(signal_to_test))
    % test for first signal's bit = 0 and second signal's bit = 1
    demod(1,n)=0;
    test=0;
    break;
end
end
end
end
end
if test
    demod(1,n)=0;
end
end
end
end

% suppression of the demodulated first signal out of the overlapped signal
bit_signal_1(1)=1;
bit_signal_1(2)=0;
bit_signal_1(3)=0;
for n=1:length(demod(1,:))
    bit_signal_1=[bit_signal_1 demod(1,n) 0];
end
bit_signal_1 = [bit_signal_1 1 0];
second_signal = signal;

former_state_1=0; former_state_2=0;
for k=1:length(bit_signal_1)
    for t=1:t_bit*F_sampling
        n=int32( (k-1)*t_bit*F_sampling+t );
        bit_1(n)=abs(bit_signal_1(k)-former_state_1)*(bit_signal_1(k)*(1-exp(-
        t*T_sampling/tau)) + former_state_1*exp(-t*T_sampling/tau)) +
        not(xor(bit_signal_1(k), former_state_1))*bit_signal_1(k);
    end
    former_state_1=bit_signal_1(k);
end
for n=1:length(bit_signal_1)
    for t=1:t_bit*F_sampling
        time=floor( (n-1)*t_bit*F_sampling+t );
        second_signal(time) = signal(time) -
        amplitude_1*sin(2*pi*freq_1*time*T_sampling+phase1) * bit_1(time);
    end
end

%DPLL_output_2=dpll(second_signal,freq_2,F_sampling, t_bit, deltaPts);
%phase2=DPLL_output_2(2,1)

% demodulation of the second signal
for n=1:13
    for t=1:t_bit*F_sampling
        time = deltaPts+3*t_bit*F_sampling+(n-1)*(2*t_bit*F_sampling)+t;
        signal_to_test(t)=second_signal(time);
theory(t)=amplitude_2*(1-exp(-t*T_sampling/tau))*sin(2*pi*freq_2*time*T_sampling+phase2);
end
for t=t_bit*F_sampling+1:2*t_bit*F_sampling
    time = deltaPts+3*t_bit*F_sampling+(n-1)*(2*t_bit*F_sampling)+t;
signal_to_test(t)=second_signal(time-1);
    theory(t)=0;
end
if sum(abs(signal_to_test-theory)) < threshold*sum(abs(theory)) \|
sum(abs(signal_to_test-shape_2)) < threshold*sum(abs(theory))
    demod(2,n)=1;
elseif max(signal_to_test)<0.2
    demod(2,n)=0;
else
    test=1;
    for delta=1:2*t_bit*F_sampling
        for t=1:delta
            test_shape_added(t)=0;
        end
        for t=delta+1:min(t_bit*F_sampling+delta,2*t_bit*F_sampling)
            time=deltaPts+3*t_bit*F_sampling+(n-1)*(2*t_bit*F_sampling)+t;
            test_shape_added(t)=amplitude_1*(1-exp(-(t-
delta)*T_sampling/tau))*sin(2*pi*freq_1*(time)*T_sampling+phase1);
        end
        for
            t=min(t_bit*F_sampling+delta,2*t_bit*F_sampling)+1:2*t_bit*F_sampling
                test_shape_added(t)=0;
        end
        if sum(abs(signal_to_test - test_shape_added - theory)) < threshold * sum(abs(signal_to_test))
            % test for first signal's bit = 1 and second signal's bit = 1 shifted
demod(2,n)=1;
test=0;
break;
end
if sum(abs(signal_to_test - test_shape_added)) < threshold * 
sum(abs(signal_to_test))
% test for first signal's bit = 0 and second signal's bit = 1 shifted
demod(2,n)=0;
test=0;
break;
end
end
end
if test
for delta=1:t_bit*F_sampling
  for t=1:delta
    time=deltaPts+3*t_bit*F_sampling+(n-1)*(2*t_bit*F_sampling)+t;
    test_shape_added(t)=amplitude_1*(1-exp(-(t-
    delta+t_bit*F_sampling)*T_sampling/tau))*sin(2*pi*freq_1*(time)*T_sampling+phase1);
  end
  for t=delta+1:2*t_bit*F_sampling
    time=deltaPts+3*t_bit*F_sampling+(n-1)*(2*t_bit*F_sampling)+t;
    test_shape(t)=amplitude_1*exp(-(t-
    delta)*T_sampling/tau)*sin(2*pi*freq_1*(time)*T_sampling+phase1);
  end
  if sum(abs(signal_to_test - test_shape_added - theory)) < 
  threshold * sum(abs(test_shape_added+theory))
    % test for first signal's bit = 1 and second signal's bit = 1
    demod(2,n)=1;
test=0;
break;
end

if sum(abs(signal_to_test - test_shape_added)) < threshold * sum(abs(test_shape_added))

    % test for first signal's bit = 0 and second signal's bit = 1
    demod(2,n)=0;
    test=0;
    break;
end
end
end
end
end
Now, here is the main program:

% Detection and process of several overlapping ASK signals

Clear all;
Close all;

% signals' bits

% Mode S reply
% the sequence 1 0 1 0 0 0 0 1 0 1 0 0 0 0 0 is the reply preambule
% reply_preambule = [ 1 0 1 0 0 0 0 1 0 1 0 0 0 0 0 ];

% Mode 3/A reply
% signal 1 A=4 B=3 C=2 D=1, ID number 4321
info_1 = [0 0 1 0 0 1 0 1 1 1 0 0 0 ]; % (C1) (A1) C2 (A2) (C4) A4 (X)
B1 D1 B2 (D2) (B4) (D4)
bit_signal_1(1)=1;
bit_signal_1(2)=0;
bit_signal_1(3)=0;
for n=1:length(info_1)
    bit_signal_1=[bit_signal_1 info_1(n) 0];
end
bit_signal_1 = [bit_signal_1 1 0 ];

% signal 2 A=4 B=3 C=2 D=2, ID number 4322
info_2 = [0 0 1 0 0 1 0 1 0 1 1 0 0 ]; % (C1) (A1) C2 (A2) (C4) A4 (X)
B1 (D1) B2 D2 (B4) (D4)
bit_signal_2(1)=1;
bit_signal_2(2)=0;
bit_signal_2(3)=0;
for n=1:length(info_2)
    bit_signal_2=[bit_signal_2 info_2(n) 0];
end
bit_signal_2 = [bit_signal_2 1 0];
for g=20:60
    for stat=1:1
        clear signal delay second_signal output_1 output_2;

        phase1=-1.2; phase2=0.8;       % signals' phases

        deltaFreq(g,stat)=90e3;
        freq_1 = 30e6;                  % frequency of the first signal
        freq_2 = freq_1+deltaFreq(g,stat); % frequency of the second signal
        F_sampling = 5*freq_1;
        T_sampling=1/F_sampling;

        t_bit=0.5e-6;                   % theoretical time for one bit
        t_bit_1 = t_bit + rand(1,1) * 0.1e-6; % effective time for one pulse in the first signal
        t_bit_2 = t_bit + rand(1,1) * 0.1e-6; % effective time for one pulse in the second signal

        a1 = 1;                        % amplitude for the first signal
        a2 = 1.5;                       % amplitude for the second signal
        tau = 0.06e-6;                  % transition time of the signals 0.05e-6

        Nb_bits_overlapped = g/3;       % number of bits overlapped

        Noise_Power = 10^(-1.5/min(a1,a2)); % not in dB
deltaT = (length(bit_signal_1)-Nb_bits_overlapped)*t_bit;       % delay between the two sequences in seconds
deltaPts = floor(deltaT*F_sampling);            % delay between the two sequences in number of points

time=[T_sampling:T_sampling:length(bit_signal_1)*t_bit + deltaT];   % time vector

length_one_signal = floor(t_bit*length(bit_signal_1)*F_sampling); % number of points for one single sequence

former_state_1=0; former_state_2=0;
for k=1:length(bit_signal_1)
    for t=1:t_bit*F_sampling
        n=int32( (k-1)*t_bit*F_sampling+t );
        bit_1(n)=abs(bit_signal_1(k)-former_state_1)*(bit_signal_1(k)*(1-
not(xor(bit_signal_1(k), former_state_1))) + former_state_1*exp(-t*T_sampling/tau)) +
        not(xor(bit_signal_1(k), former_state_1))*bit_signal_1(k);
        bit_2(n)=abs(bit_signal_2(k)-former_state_2)*(bit_signal_2(k)*(1-
not(xor(bit_signal_2(k), former_state_2))) + former_state_2*exp(-t*T_sampling/tau)) +
        not(xor(bit_signal_2(k), former_state_2))*bit_signal_2(k);
    end
    former_state_1=bit_signal_1(k);
    former_state_2=bit_signal_2(k);
end
if length(bit_2)<length(time)-deltaPts
    bit_2(length(time)-deltaPts)=0;
    bit_1(length_one_signal)=0;
end
if (deltaPts<length_one_signal)
    for t=1:deltaPts
        signal(t) = a1 * bit_1(t) * sin(2*pi*freq_1*t*T_sampling+phase1);
end
for t=deltaPts+1:length_one_signal
    signal(t) = a1 * bit_1(t) * sin(2*pi*freq_1*t*T_sampling+phase1) +
                a2 * bit_2(t-deltaPts) * sin(2*pi*freq_2*t*T_sampling+phase2);
end
for t=length_one_signal+1:length(time)
    signal(t) = a2 * bit_2(t-deltaPts) *
                sin(2*pi*freq_2*t*T_sampling+phase2);
end
else
    for t=1:length_one_signal
        signal(t) = a1 * bit_1(t) * sin(2*pi*freq_1*t*T_sampling+phase1);
    end
    for t=length_one_signal+1:deltaPts
        signal(t) = 0;
    end
    for t=deltaPts+1:length(time)
        signal(t) = a2 * bit_2(t-deltaPts) *
                    sin(2*pi*freq_2*t*T_sampling+phase2);
    end
end
signal = signal + Noise_Power * randn(1,length(signal) );

% detection of the first signal
DPLL_output_1 = dpll(signal,freq_1,F_sampling, t_bit, 1);
PLLphase1 = DPLL_output_1(2,1)
error_1 = DPLL_output_1(3,1);
if error_1
    disp('PLL unable to lock on the first signal');
end
DPLL_output_2=dpll(signal,freq_2,F_sampling, t_bit, deltaPts+29*t_bit*F_sampling);
PLLphase2  = DPLL_output_2(2,1)
error_2 = DPLL_output_2(3,1);
if error_2
    disp(' ')
    disp('PLL unable to lock on the second signal');
end

for t=1:2*t_bit*F_sampling
    shape_1(t) = signal(t);
    shape_2(t) = signal(deltaPts+29*t_bit*F_sampling+t-1);
end

demod_comparator(g,:,:) = comparator (signal, deltaPts, t_bit, F_sampling, freq_1, freq_2, PLLphase1, PLLphase2, shape_1, shape_2);
for n=1:length(info_1)
    demod_comparator_1(n)=demod_comparator(g,1,n);
    demod_comparator_2(n)=demod_comparator(g,2,n);
end
BER_comparator_1(g)=sum(abs(demod_comparator_1 - info_1));
BER_comparator_2(g)=sum(abs(demod_comparator_2 - info_2));

% second demodulation
% suppression of the demodulated second signal out of the overlapped signal
clear reconstructed_bit_signal_2;
amplitude_2=a2;  % we can detect the exact amplitude analogically
reconstructed_bit_signal_2(1)=1;
reconstructed_bit_signal_2(2)=0;
reconstructed_bit_signal_2(3)=0;
for n=1:length(info_2)
    reconstructed_bit_signal_2=[reconstructed_bit_signal_2
demod_comparator_2(n) 0];
end
reconstructed_bit_signal_2 = [reconstructed_bit_signal_2 1 0];
first_signal = signal;

former_state_1=0; former_state_2=0;
for k=1:length(reconstructed_bit_signal_2)
    for t=1:t_bit*F_sampling
        n=floor( deltaPts+(k-1)*t_bit*F_sampling+t);
        reconstructed_bit_2(n)=abs(reconstructed_bit_signal_2(k)-
former_state_2)*(reconstructed_bit_signal_2(k)*(1-exp(-t*T_sampling/tau)) +
former_state_2*exp(-t*T_sampling/tau)) + not(xor(reconstructed_bit_signal_2(k),
former_state_2))*reconstructed_bit_signal_2(k);
    end
    former_state_2=reconstructed_bit_signal_2(k);
end
for n=1:length(reconstructed_bit_signal_2)
    for t=1:t_bit*F_sampling
        time=floor( deltaPts + (n-1)*t_bit*F_sampling+t);
        first_signal(time) = signal(time) -
amplitude_2*sin(2*pi*freq_2*time*T_sampling+phase2) *
reconstructed_bit_2(time);
    end
end

% suppression of the demodulated first signal out of the overlapped
signal
clear reconstructed_bit_signal_1;
amplitude_1 = a1;
reconstructed_bit_signal_1(1)=1;
reconstructed_bit_signal_1(2)=0;
reconstructed_bit_signal_1(3)=0;
for n=1:length(info_1)
    reconstructed_bit_signal_1=[reconstructed_bit_signal_1
demod_comparator_1(n) 0];
end
reconstructed_bit_signal_1 = [reconstructed_bit_signal_1 1 0];
second_signal = signal;

former_state_1=0; former_state_2=0;
for k=1:length(reconstructed_bit_signal_1)
    for t=1:t_bit*F_sampling
        n=floor((k-1)*t_bit*F_sampling+t);
        reconstructed_bit_1(n)=abs(reconstructed_bit_signal_1(k)-
former_state_1)*(reconstructed_bit_signal_1(k)*(1-exp(-t*T_sampling/tau)) +
former_state_1*exp(-t*T_sampling/tau)) + not(xor(reconstructed_bit_signal_1(k),
former_state_1))*reconstructed_bit_signal_1(k);
    end
    former_state_1=reconstructed_bit_signal_1(k);
end
for n=1:length(reconstructed_bit_signal_1)
    for t=1:t_bit*F_sampling
        time=floor((n-1)*t_bit*F_sampling+t);
        second_signal(time) = signal(time) -
amplitude_1*sin(2*pi*freq_1*time*T_sampling+phase1) *
reconstructed_bit_1(time);
    end
end
second_demodulation_1(g,:,:) = comparator(first_signal, deltaPts, t_bit, 
F_sampling, freq_1, freq_2, PLLphase1, PLLphase2, shape_1, shape_2);
second_demodulation_2(g,:,:) = comparator (second_signal, deltaPts, 
t_bit, F_sampling, freq_1, freq_2, PLLphase1, PLLphase2, shape_1, shape_2);

for n=1:length(info_1)
    demod_comparator_1(n)=second_demodulation_1(g,1,n);
    demod_comparator_2(n)=second_demodulation_2(g,2,n);
end
BER_comparator_second_1(g)=sum(abs(demod_comparator_1 - 
info_1));
BER_comparator_second_2(g)=sum(abs(demod_comparator_2 - 
info_2));
end
end
Appendix B: Mathematical Analysis

In these appendixes, the reader will find some help for a better understanding of Phase Lock Loop (PLL) basic principle, and Amplitude Shift Key (ASK) demodulation. Finally, a review of different transponders modes and Traffic Collision Avoidance System (TCAS) is presented.

B.1 Analysis of PLL output

Phase-Locked Loops are a coherent detection method widely used in telecommunications. They are now principally used for all communication systems where path loss and power consumption are critical. After reviewing the way a PLL works and how it acquires both the input signal frequency and phase, we will review the PLL transfer function, the PLL damping factor, and PLL bandwidth.

B.1.1 Generalities about PLL

The following block diagram can describe the general form of a PLL:

![General PLL block diagram](image)

Figure B-1 General PLL block diagram
The two signals $v_1$ and $v_2$ are applied to a phase detector. The result of this operation is then transmitted to a Low Pass Filter (LPF). The LPF output $v_e$ is called the error function of the loop, and is equal to zero when $v_1$ and $v_2$ are identical. The signal $v_e$ is used as a command signal for the sinusoidal characteristic Voltage Controlled Oscillator (VCO). Now, let us study how a PLL locks.

For an input signal $v_1$ and a feedback signal $v_2$ that are sin waves

\[
v_1(t) = \sin(2\pi f_1 t + \phi_1) \quad \text{(Eq. B.1)}
\]

\[
v_2(t) = \cos(2\pi f_2 t + \phi_2) \quad \text{(Eq. B.2)}
\]

\[
v_0(t) = K_d \times \sin(2\pi (f_1 - f_2) t + \phi_1 - \phi_2) \quad \text{(Eq. B.3)}
\]

The VCO output instantaneous frequency is a linear function of the command voltage $v_e$ around a central frequency $f_2$

\[
\frac{\partial}{\partial t} \phi_2 = K_v \times v_e \quad \text{(Eq. B.4)}
\]

Therefore, provided that the frequency difference between $f_1$ and $f_2$ is within the loop lock range to be determined later, $v_0$ can be written in a different way:

\[
v_0(t) = K_d \sin(\phi_1 - \phi(t)) \quad \text{(Eq. B.5)}
\]

Where $\phi$ is a notation to describe the VCO phase corresponding to $\phi_2$ and is equal to:

\[
\phi(t) = \phi_2 - 2\pi (f_1 - f_2) t \quad \text{(Eq. B.6)}
\]

Providing that the frequency difference between $f_1$ and $f_2$ is within the loop lock range, $\phi(t)$ is actually a DC signal. Therefore, $v_e$ is equal to $v_0$ and the VCO instantaneous frequency is

\[
2\pi f_{\text{inst}}(t) = 2\pi f_2 + K_v \times v_0(t) \quad \text{(Eq. B.7)}
\]

Thus, from Eq. B.2

\[
\cos(2\pi f_{\text{inst}} t + \phi) = \cos(2\pi f_2 t + \phi_2) \quad \text{(Eq. B.8)}
\]

and therefore (from Eq. B.6 and B.7)

\[
2\pi f_2 + K_v \times v_0(t) + \phi_2 - 2\pi (f_1 - f_2) t = 2\pi f_2 t + \phi_2 \quad \text{(Eq. B.9)}
\]
from which we conclude

\[ v_0(t) = \frac{1}{K_v} \times (f_1 - f_2) \]  
\text{(Eq. B.10)}

or noting \( \Delta f \) the frequency difference between \( f_1 \) and the starting VCO frequency \( f_2 \)

\[ v_0(t) = v_e(t) = \frac{1}{K_v} \times \Delta f \]  
\text{(Eq. B.11)}

Let us examine now the differential equations we have obtained. To simplify the problem, we will replace the Low Pass Filter by an amplifier with a coefficient \( K_a \). Equation B.4 becomes now

\[ \frac{\partial}{\partial t} \phi_2 = K_v \times K_a \times v_0 \]  
\text{(Eq. B.12)}

From Equations B.4 and B.12 we obtain

\[ \frac{\partial}{\partial t} \phi_2 = K_v \times K_a \times K_d \times \sin(2\pi (f_1 - f_2) t + \phi_1 - \phi_2) \]  
\text{(Eq. B.13)}

Stating \( \Phi_0 = 2\pi \Delta f t + \phi_1 - \phi_2 \) and \( K = K_v \times K_a \times K_d \)

\[ \frac{\partial}{\partial t} \Phi_0 = \Delta f - \frac{\partial}{\partial t} \phi_2 \]  
\text{(Eq. B.14)}

Combining Equations B.13 and B.14, we obtain

\[ \frac{\partial}{\partial t} \Phi_0 = \Delta f - K \cos \Phi_0 \]  
\text{(Eq. B.15)}

or

\[ \frac{\partial t}{\partial t} = \frac{\partial \Phi_0}{(\Delta f - K \cos \Phi_0)} \]  
\text{(Eq. B.16)}

Integrating Equation B.16 between the present time \( t \) and the phase reference time \( t_0 \), we obtain two results, depending upon the relative value of \( \Delta f \) and \( K \).

Case (a) : \( |\Delta f| > |K| \)

\[ t - t_0 = \frac{2}{\sqrt{\Delta f^2 - K^2}} \arctan \left\{ \left( \frac{\Delta f + K}{\sqrt{\Delta f^2 - K^2}} \right) \right\} \]

thus

\[ \Phi_0(t) = 2 \arctan \left\{ \frac{\sqrt{\Delta f^2 - K^2}}{\Delta f + K} \tan \left( \frac{1}{2} \sqrt{\Delta f^2 - K^2} \times (t - t_0) \right) \right\} \]

This solution does not have a limit for \( t \rightarrow \infty \) Therefore, the VCO output does not have a constant phase (the PLL is said to be out of lock), and \( \phi_2 \) keeps oscillating with a period \( T \)
\[ T = \frac{2\pi}{\sqrt{\Delta f^2 - K^2}} \]

When \( \Delta f \to K \), the period \( T \to \infty \) and the VCO output is equal to a constant after a certain period of time.

Case (a) : \(|\Delta f| > |K|\)

In that case Equation B.16 cannot be solved analytically [Blanchard, 1976]. Let us see how the PLL behaves in that case.

The voltage \( v_0 \) (and then \( v_0 \)) becomes a DC signal, and the VCO output becomes synchronous with the input signal \( v_1 \). The VCO output instantaneous frequency is equal to the sum of \( f_2 \) and the derivative of \( \phi \) with respect to time. From Equation B.7

\[ 2\pi f_{\text{inst}}(t) = \partial / \partial t \left( 2\pi f_2 t + \varphi \right) = 2\pi f_2 + K_v \times v_0(t) \]  

(Eq. B.17)

Therefore,

\[ \partial / \partial t \left( \varphi \right) = K_v \times v_0(t) = K_v \times K_d \times \sin(\phi_1 - \varphi) \]  

(Eq. B.18)

Thus,

\[ \phi(t) = \phi_1 - \arcsin \left\{ \frac{2\pi f_1 - 2\pi f_2}{K_v \times K_d} \right\} \]  

(Eq. B.19)

And then from Equation B.5

\[ v_0 = 2\pi \times \Delta f / K_v \]

Finally, from Equation B.17

\[ 2\pi f_{\text{inst}}(t) = 2\pi f_2 + K_v \times \Delta f \times 2\pi / K_v = 2\pi f_1 \]  

(Eq. B.20)

Now, the two frequencies are equal, and the VCO can lock on the input signal, \( v_1 \). From Equation B.16,

\[ \partial t = \partial \Phi / (-K \cos \Phi) \]  

(Eq. B.21)

which gives

\[ -K \left( t - t_0 \right) = \frac{1}{2} \ln \left\{ \frac{1 + \sin \Phi_0}{1 - \sin \Phi_0} \right\} \]  

(Eq. B.22)

When \( t \) increases, the term in the logarithm goes toward zero. Therefore \( \Phi_0 \to -\pi/2 \). The PLL is said to be locked. The VCO output is equal to \( \phi_1 + \pi/2 \).
B.1.2 Bandwidth and Damping factor

For a LPF transfer function \( G(s) \), the PLL linear transfer function is [Smith, 1997]

\[
\phi_2 / \phi_1 = G(s) / \{ 1 + G(s) \} \tag{Eq. B.23}
\]

Let us study the case of an order-one low pass filter, with a transfer function equal to

\[
G(s) = \omega_c / \{ \omega_c + s \} \tag{Eq. B.24}
\]

We quickly find the transfer function

\[
\phi_2 / \phi_1 = K / \{ s ( s / \omega_c + 1 ) + K \} \tag{Eq. B.25}
\]

\[
\phi_2 / \phi_1 = 1 / \{ s^2 / \omega_p^2 + 2 \times \xi \times s / \omega_p + 1 \} \tag{Eq. B.26}
\]

where the damping factor \( \xi \) and the PLL cut-off frequency are equal to

\[
\omega_p = K \times \omega_c \tag{Eq. B.27}
\]

\[
\xi = 1/2 \omega_p / K \tag{Eq. B.28}
\]

For the frequency equal to the 3-dB bandwidth, the magnitude of the transfer function given by Equation B.26 is equal to -3dB. Solving this equation, we find the PLL 3-dB bandwidth [Smith, 1997]

\[
\omega_{pll} = \omega_p \times \{ 1 - 2 \xi^2 + ( 2 - 4 \xi^2 + 4 \xi^4 )^{1/2} \}^{1/2} \tag{Eq. B.29}
\]

Now, let us determine the noise equivalent bandwidth \( B_n \) for a PLL input noise power of \( n_0 \). A noise added to the PLL input signal results in a noise of the VCO output phase. The standard deviation \( \sigma \) of the VCO phase is defined as

\[
\sigma = \frac{n_0 \times \omega_p}{A^2} \int \left| \frac{\phi_2}{\phi_1} \right| (2\pi f)^2 df \tag{Eq. B.30}
\]

where \( A \) is the PLL input signal's amplitude, and the integral boundaries are infinite. From Equation B.26, we obtain

\[
\sigma = \frac{n_0 \times \omega_p}{A^2 \times 4 \xi} \tag{Eq. B.31}
\]

Defining Signal-to-Noise Ratio for the VCO output phase as

\[
SNR = \frac{A^2}{2 \times n_0 \times B_n} \tag{Eq. B.32}
\]
We obtain the equivalent noise bandwidth
\[ B_n = \frac{\omega_p}{2\xi} \]  
(Eq. B.33)

**B.1.3 Sum of two signals**

Let us study how a VCO behaves when the PLL input signal is equal to the sum of two sinusoids. With a linear approximation that we can make when the PLL is nearly locked on, if the PLL input is the sum of two sinusoids, the PLL output is the sum of the two sinusoids’ PLL responses.

Therefore, if the two input signals have the same frequency then the PLL will lock on with phase \( \frac{1}{2}(\phi_1 + \phi_2) \). Otherwise, it will not lock on, since \( f_{vco} \) cannot be equal to both frequencies \( f_1 \) and \( f_2 \) at the same time.

**B.2 Analysis of the transponders responses separation**

This appendix explains how the system proceeds to recover the bits from an Amplitude Shift Key (ASK) signal, and then how to separate two overlapped ASK signals using the system described in this thesis (Chapter 3). We will see when two overlapped signals cannot be recovered by a simple ASK demodulation.

**B.2.1 System general output for a sinusoid**

Let us consider the following system block diagram:

![Demodulator block diagram](image)

**Figure B-2** Demodulator block diagram
This figure reviews the system described in Chapter 3 and used to demodulate the transponders responses. We will assume that the rebuilder output is a sinusoid whose phase is equal to the input signal phase and whose frequency is equal to the input signal frequency.

If the input consists in a single ASK signal:

\[ X(t) = a(t).\sin(2\pi f_0 t + \phi_0) \quad \text{(Eq. B.34)} \]

where \( a(t) \in \{1,0\} \) depending on the bit value

Then, the multiplier output is equal to

\[ Y(t) = a(t).\sin(2\pi f_0 t + \phi_0).\sin(2\pi f_0 t + \phi_0) \quad \text{(Eq. B.35)} \]

\[ Y(t) = a(t)/2.\{ \cos(0) - \cos(2\pi 2f_0 t + 2\phi_0) \} \quad \text{(Eq. B.36)} \]

Thus, the LPF output is

\[ Z(t) = a(t)/2 \quad \text{(Eq. B.37)} \]

Therefore, the demodulation is straightforward.

### B.2.2 Two ASK overlapped signals

Now, for two overlapped ASK signals, demodulated using \( f_a \) as frequency reference (the rebuilder uses \( f_a \) as frequency reference)

\[ X(t) = a(t).\sin(2\pi f_a t + \phi_a) + b(t).\sin(2\pi f_b t + \phi_b) \quad \text{(Eq. B.38)} \]

\[ Y(t) = a(t)/2.\{ 1 - \cos(2\pi 2f_a t + 2\phi_a) \} + b(t)/2.\{ \cos(2\pi( f_b - f_a) t + \phi_b - \phi_a) - \cos(2\pi( f_a + f_b) t + \phi_b) \} \quad \text{(Eq. B.39)} \]

Now, one can distinguish two cases:

1) \( f_a = f_b \)  \( Z(t) = \frac{1}{2} \{ a(t)+b(t) \} \) and no separation is possible directly. We can still show (Stevens, Honold) that separation is still possible if and only if the two signals are not synchronized (the bits are not totally overlapped).

2) \( f_a \neq f_b \) \( Z(t) = \frac{1}{2} a(t) + \frac{1}{2} b(t) \cos(2\pi ( f_b - f_a) t + \phi_b - \phi_a) \)
If the frequency and phase difference are low enough for the cosine term to be considered constant, one can still separate $a(t)$ from $b(t)$. Otherwise, the cosine term may affect the output. (see figure 3.12 to view an example where the cosine term is visible for the second signal $b(t)$)

**B.3 Transponders modes formats**

General aviation transponders are divided into different modes. The modes used in this thesis are Mode A, Mode C and Mode S.

**B.3.1 Interrogation Format**

All interrogation formats are composed by three pulses, called P1, P2 and P3. The spacing between P1 and P3 determines which mode is used, as shown on the following table:

<table>
<thead>
<tr>
<th>Mode</th>
<th>Pulse spacing (µs)</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>Military IFF</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>Military Individual Code</td>
</tr>
<tr>
<td>3/A</td>
<td>8</td>
<td>Military ATC / IFF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Civil ATC</td>
</tr>
<tr>
<td>B</td>
<td>17</td>
<td>Civil ATC</td>
</tr>
<tr>
<td>C</td>
<td>21</td>
<td>ATC / altitude transmission</td>
</tr>
<tr>
<td>D</td>
<td>25</td>
<td>Civil ATC (not yet in use)</td>
</tr>
</tbody>
</table>

*Figure B-3* Interrogation formats table

**B.3.2 Response Format**
Two pulses F1 and F2 indicating the beginning and the end of the sequence, and 12 information pulses compose the response format. There is also a pulse X called empty in the middle of the sequence.

![Diagram of Mode A transponder response format]

**Figure B-4** General response format from a Mode A transponder

The information pulses are divided into four groups of three pulses (A, B, C and D). Within each group, each pulse has a specific numerical value. For instance A1 = 1000, A2 = 2000 and A4 = 4000. In the example presented above, the coded value is

\[4000 + 300 + 20 + 1 = 4321.\]

To these two modes A and C, we need to add the Mode S. Mode S is used like other modes except the response contains both identity and altitude.

**B.4 TCAS**

Traffic Collision Avoidance System is an on-board system that uses Mode S to detect potential traffic conflict and compute an escape maneuver [Williamson & Spencer, 1989]. TCAS works only when Air Traffic Control has failed; it is an
emergency device. Three series of TCAS have been developed: TCAS I, TCAS II and TCAS III [Stevens, 1988].

TCAS I was the first system introduced, and has now been developed for small aircraft. The Mode S transponder detects a potential conflict at the same flight level (same altitude) and gives approximately the direction of the conflict. The pilot needs to detect visually the other aircraft and to decide the escape maneuver.

TCAS II tracks nearby aircraft horizontally and vertically. The time and location of impact are computed and, in certain circumstances, the pilot is advised a solution. If the threat is also equipped with a TCAS system, the escape maneuver is coordinated between the two aircraft.

TCAS III is very similar to TCAS II; the antenna is simply more accurate and a horizontal escape solution can be provided.

The recent air crash in Switzerland (July 1st 2002) occurred because of a failure in the Air Traffic Control (ATC) system. The conflict was detected too late by the ATC: the pilots and the controller did not have time to coordinate. One pilot followed the TCAS instructions (and that is the correct procedure) while the other one followed the controller instructions. Moreover, the controller has not been advised of the TCAS solution.
References


Vita

Charles Florin was born in Rouen, France on October 27, 1979. He was raised in Rouen until the age of 17. Upon graduation from Lycee Saint Jean-Baptiste de La Salle, Charles attended Lycee Stanislas, Paris to prepare the competitive exam for the entrance to Grandes Ecoles. Among the different majors available, he chose Mathematics.

After passing this exam, he was admitted to Supelec, Paris, where he has been participating in different student associations: he was a representative of students and the chief redactor of Supelec's newspaper.

After his studies in France, Charles has attended Virginia Polytechnic Institute and State University, where he specialized in communication engineering, and specially radar systems, digital and satellite communication.